

Reduction of the total execution time to achieve the optimal k -node reliability of distributed computing systems using a novel heuristic algorithm

Chin-Ching Chiu^a, Yi-Shiung Yeh^{a,*}, Ruey-Shun Chen^b

^a*Institute of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan*

^b*Institute of Information Management, Institute of Computer Science and Information Engineering, Hsinchu, Taiwan*

Received 4 January 1999; received in revised form 4 June 1999; accepted 4 August 1999

Abstract

A distributed computing system is a collection of processor–memory pairs connected by communication links. A k -node set is a subset of total nodes in a distributed computing system. A k -node set with capacity constraint is a k -node set that possesses sufficient node capacity. Because computing the reliability of a distributed computing system is generally an NP-hard problem, an adequate k -node set with a given capacity constraint must be determined by an effective algorithm with an approximate reliability. Relatively few investigations, namely an exact method and a k -tree reduction method, have examined k -node reliability optimization with capacity constraint. Such investigations either spent an exponential time or rarely obtained an optimal solution. Therefore, in this work, we present a novel heuristic algorithm to reduce the computational time and deviation from an exact solution. The proposed algorithm has simple independent steps, including selection of k -node sets according to a node's weight or a link's weight. The number of selected k -node sets is either one or two, thereby spending less time to compute the reliability of k -node sets. Computational results demonstrate that the proposed algorithm is more effective and provides a better solution for a large distributed computing system than those in previous investigations. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Distributed computing systems (DCS); Reliability optimization; k -Node reliability (KNR)

1. Introduction

Recent developments in computer networking and low cost computational elements have led to increasing interest in distributed computing systems (DCS). DCS, a collection of processor–memory pairs connected by communication links, is logically integrated by a distributed communication network. The communication subnet may be a geographically dispersed collection of communication processors or a local area network [1–3]. The numerous merits of using DCS include improved resource sharing, enhanced fault tolerance and high reliability.

The economic benefits of resource sharing largely account for the importance of DCS. A DCS focuses on providing efficient communication among various nodes, thereby increasing their reliability and making their service available to more users [4]. Designing such systems must consider system reliability which heavily relies on the topological layout of communication links [5–7].

The topology of a network can be characterized by a linear graph. These network topologies can be characterized by their network reliability, message-delay, or network capacity. The performance characteristics depend on many properties of the network topology [8–13]: the number of ports at each node (degree of a node) and the number of links. The number of links directly impacts the system reliability.

Previous literature provides reliability optimization models of DCS that optimize source to destination reliability, k -out-of- n systems reliability and overall system reliability [7,14,15]. Two reliability optimization models have been presented in [16].

As defined, k -node reliability is the probability that nodes in the k -node set (subset of the set of processing elements) in the DCS are connected. The exact method (EM) [17] and the k -tree reduction method [18] have examined k -node reliability optimization with capacity constraint. These either spent an exponential time or barely obtained an optimal solution.

This work focuses mainly on how to compute nearly maximum system reliability objectives with capacity

* Corresponding author. Tel.: +886-3-5731813; fax: +886-3-5724176.
E-mail address: ysyeh@csie.natu.edu.tw (Y.-S. Yeh)

constraint. The fact that computing the reliability of DCS is a NP-hard problem explains why an adequate solution must be derived in a very short time. More specifically, deriving a solution with an exact solution as possible is of primary concern. Therefore, this work computes the reliability of a subset of the set of processing elements such that the reliability is maximized and the specified capacity constraint is satisfied.

2. Problem description

In this section, we described the problem addressed herein to clarify our research objectives.

2.1. Notations and definitions

The following notations and definitions will be used.

$c(G_k)$, $c(G'_k)$	the sum of capacity of the k -node set G_k and G'_k , respectively
$c(v_i)$	the capacity of the i^{th} node.
$d(v_i)$	the number of links connected to the node v_i
e	the number of links in G , $e = E $
e_{ij}	an edge represents a communication link between v_i and v_j , $e_{ij} \in E$
n	the number of nodes in G , $n = V $.
p_{ij}	the probability of success of link e_{ij} .
q_{ij}	the probability of failure of link e_{ij} .
v_i	the i^{th} processing element or the i^{th} node, $v_i \in V$
$w(v_i)$	the weight of the i^{th} node
$w(e_{ij})$	the weight of link e_{ij}
y_{ij}	the number of path which length is 2 between v_i and v_j
$C_{\text{constraint}}$	total capacity limit in a DCS.
$G = (V, E)$	an undirected DCS graph where V denotes a set of processing elements, and E represents a set of communication links
G_k, G'_K	the graph G with the set K of nodes specified, and $ K \geq 2$, $G_k, G'_K \subset G$
P	the link reliability matrix where $P(i, j) = P(j, i) = p_{i,j}$ if $e_{i,j}$ exists in G , $p(i, j) = p(j, i) = 0$, otherwise for $i, j = 1, 2, \dots, n$
$R(G_k)$,	the reliability of the k -node set
$R(G'_K)$	solution of a DCS graph G
$V_{\text{adj}(G_k)}$, $V_{\text{adj}(G'_k)}$	a set of nodes which are adjacent to any node of G_k and G'_K , respectively

Definition 1. A k -node reliability(KNR) is defined as the probability that a specified set K of nodes is connected (where K denotes a subset of the set of processing elements).

Definition 2. A node v_i is directly connected to a set G_k of

nodes if and only if there is a link between v_i and a node in G_k .

Definition 3. The number of reliability computation (NRC) is the number of k -node sets whose reliability should be computed.

2.2. Problem statements

Bi-directional communication channels operate between processing elements. A distributed computing system can be modeled by a simple undirected graph. For a topology of the DCS with four nodes, say $V = \{v_1, v_2, v_3, v_4\}$, and five links, say $E = \{e_{1,2}, e_{1,3}, e_{2,3}, e_{2,4}, e_{3,4}\}$, there are many subsets of nodes. A set, K , is a subset of the DCS which includes some nodes of the given node set V . The KNR is the probability that a specified set K of nodes is connected, where K denotes a subset of set of processing elements. For example, $K = \{v_1, v_2, v_3\}$ is selected in the DCS with bridge topology. The reliability of the set, K , can be computed by means of a sum of mutually disjoint terms [19].

$$R(G_k) = p_{1,2}p_{1,3}q_{2,3}q_{2,4} + p_{1,2}p_{1,3}q_{2,3}p_{2,4}q_{3,4} + p_{1,2}q_{1,3}p_{2,3}q_{2,4} \\ + p_{1,2}q_{1,3}p_{2,3}p_{2,4}q_{3,4} + p_{1,3}p_{2,3}q_{2,4} + p_{1,3}p_{2,3}p_{2,4}q_{3,4} \\ + p_{1,2}q_{1,3}p_{2,4}p_{3,4} + p_{1,3}p_{2,4}p_{3,4}$$

Assume that probability of $p_{1,2}, p_{1,3}, p_{2,3}, p_{3,4}$ is 0.95, 0.94, 0.93, 0.92, and 0.91, respectively. Then $R(G_k) = 0.9958148$.

A k -node reliability problem can be characterized as follows:

Given

Topology of a DCS.

The reliability of each communication link.

The capacity of each node.

A set of data files.

Assumption

Each node is perfectly reliable.

Each link is either in the working (ON) state or failed (OFF) state.

Constraint

The total capacity of data files to be allocated.

Goal

To select a specified set K of nodes in a DCS to which to allocate data files, by doing so, k -node reliability is adequate under capacity constraint.

Reliability optimization can be defined in the maximum reliability for computing a given task under some constraints. For a given task, its reliability can be computed as R_1, R_2, \dots, R_x for x conditions. By doing so, the reliability optimization for the task is the maximal reliability in R_1, R_2, \dots, R_x . The heuristic algorithm involves obtaining

an approximate solution which is close to the maximal reliability in R_1, R_2, \dots, R_x . Restated, a set K of nodes is to be found from the given node set V of a DCS such that the k -node set reliability is adequate and the total capacity satisfies the capacity constraint. The main problem can be mathematically stated as follows:

Object: Maximize $R(G_k)$
 subject to: $\sum_{v_i \in G_k} c(v_i) \geq C_{\text{constraint}}$,
 where $R(G_k)$, $c(v_i)$, $C_{\text{constraint}}$ are defined in Section 2.1.

Obviously, the problem for a large DCS such as a metropolitan area network requires a large execution time. Herein, we develop an effective method that allows k -node reliability optimization in a DCS achieve the desired performance. Owing to its computational advantages, a proposed method may be preferred to the EM and the k -tree reduction method when a DCS is large.

3. Heuristic algorithm for k -node reliability

In this section, we present a heuristic algorithm to maximize k -node reliability. The analysis performed herein assumes that all nodes are perfect and links are unreliable.

3.1. The concept of the proposed algorithm

The EM, an optimal solution, requires excessive execution time in a large DCS and cannot effectively reduce the problem space. Occasionally, an application requires an efficient algorithm to compute the reliability due to its resource considerations. In these circumstances, achieving optimal reliability may not be desired. Instead, an effective algorithm with an approximate reliability is highly attractive. In fact, most DCS are large and an increasing number of nodes causes exponential growth of the execution time for a solution. Although able to reduce computational time, the k -tree reduction method has much difficulty in deriving the optimal solution. Therefore, this work presents a novel algorithm to reduce the total execution time to achieve the optimal KNR of DCS.

Consider a DCS with n nodes and e links. The capacity constraint is $C_{\text{constraint}}$, where its optimal DCS topology is the set K of nodes. Restated, the set K of nodes has the maximum reliability and its total capacity is at least as large as the capacity limit $C_{\text{constraint}}$.

The reliability of a k -node set is dependence on the number of links which incident to a node of the k -node set and each link reliability. For any node, the degree of that node affecting the number of paths of information can be transferred from others' nodes. Therefore, we employ a relatively simple means of computing the node value, which takes less time and can quickly compute the weight of each node. The following formula is used to compute the

weight of node v_i .

$$w(v_i) = p_{i,k_1} + \sum_{x=2}^{d(v_i)} \left(\prod_{z=1}^{x-1} q_{i,k_z} \right) p_{i,k_x}. \quad (1)$$

The above formula can be rewritten as follows:

$$w(v_i) = p_{i,k_1} + q_{i,k_1} * (p_{i,k_2} + (q_{i,k_2} * (p_{i,k_3} + (\dots (q_{i,k_{d(v_i)-1}} * p_{i,k_{d(v_i)}})) \dots))). \quad (2)$$

The above formula is easy to program and much reduces the number of multiplication. If the degree of v_i is $d(v_i)$, the weight of v_i can be computed in $d(v_i) - 1$ additions and $d(v_i) - 1$ multiplication. Thus, we can obtain the weight of every node in $2e$ additions and $2e$ multiplication.

In the network, two nodes may contain many paths between them. A path's length is between one and $n - 1$. For reducing the computational time, we consider the path in which length is not greater than two. The following formula is used to evaluate the weight of link e_{ij} .

$$w(e_{ij}) = p_{ij} + q_{ij} p_{i,k_1} p_{k_1,j} + \sum_{x=2}^{y_{ij}} \left(\prod_{z=1}^{x-1} (1 - p_{i,k_z} p_{k_z,j}) \right) p_{i,k_x} p_{k_x,j}. \quad (3)$$

The above formula can be rewritten as follows:

$$w(e_{ij}) = p_{ij} + q_{ij} * (p_{i,k_1} p_{k_1,j} + (1 - p_{i,k_1} p_{k_1,j}) * (p_{i,k_2} p_{k_2,j} + (1 - p_{i,k_2} p_{k_2,j}) * (\dots ((1 - p_{i,k_{y_{ij}-1}} p_{k_{y_{ij}-1},j}) p_{i,k_{y_{ij}}} p_{k_{y_{ij}},j}))) \dots)). \quad (4)$$

The formula can easily edit a program and much reduces the number of multiplicative operations. Where y_{ij} denotes the number in which the length of a path between v_i are v_j is two. In addition, the value of y_{ij} is not greater than $n - 2$. The weight of e_{ij} can be computed in y_{ij} additions, y_{ij} subtractions and $3y_{ij}$ multiplication. Thus, in the worst case, when the graph is a complete one, all the weights of each link in $e(n - 2)$ additions, $e(n - 2)$ subtractions and $3e(n - 2)$ multiplication can be obtained. Using the characteristic of side effect of programming language, we can finish in $2e(n - 2)$ multiplication instead of $3e(n - 2)$ multiplication.

In this work, we propose a novel heuristic algorithm to compute k -node reliability. The algorithm has simple independent steps, including selection of k -node sets according to a node's weight or link's weight. The following observations can be made on how to reduce a k -node set. For a given selected k -node set, the reliability of this k -node set is less than the reliability of its subset. Thus, during the reliability evaluation process, if a subset of the k -node set satisfies the capacity constraint, then the k -node set should be replaced by its subset. After obtaining these k -node sets, we compute their reliability and output the k -node set of maximal reliability.

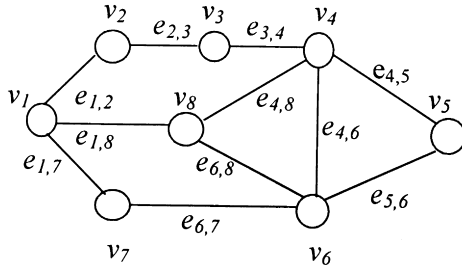


Fig. 1. The DCS with eight nodes and eleven links.

$$\begin{array}{cccc}
 c(v_1)=39 & c(v_2)=45 & c(v_3)=38 & c(v_4)=53 \\
 c(v_5)=47 & c(v_6)=49 & c(v_7)=51 & c(v_8)=41 \\
 p_{1,2}=0.89 & p_{1,7}=0.81 & p_{1,8}=0.93 & p_{2,3}=0.85 \\
 p_{2,4}=0.91 & p_{4,5}=0.82 & p_{4,6}=0.83 & p_{4,8}=0.96 \\
 p_{5,6}=0.87 & p_{6,7}=0.84 & p_{6,8}=0.88 & \\
 C_{\text{constraint}} \geq 100
 \end{array}$$

3.2. The proposed heuristic algorithm

In the following, we present a heuristic algorithm to maximize k -node reliability optimal design of a DCS under capacity constraint.

Algorithm KNR

step 0 Initializing, reading system parameters: n , e , $C_{\text{constraint}}$, P , $c(v_i)$, $i = 1, \dots, n$.

step 1 /*Evaluating the weight of each node by formula (2).*/

Let $E_{\text{tmp}} = E$.

dowhile($E_{\text{tmp}} \neq \emptyset$)

 choose a link, say $e_{i,j}$, from E_{tmp} .

 if ($w(v_i) = 0$)

$w(v_i) = p_{i,j}$.

 else

$w(v_i) = p_{i,j} + q_{i,j} * w(v_i)$.

 end if

 if ($w(v_j) = 0$)

$w(v_j) = p_{i,j}$

 else

$w(v_j) = p_{i,j} + q_{i,j} * w(v_j)$.

 end if

 Let $E_{\text{tmp}} = E_{\text{tmp}} - \{e_{i,j}\}$ /* discard $e_{i,j}$ from E_{tmp} */

end dowhile

Sorting all nodes according to their weight in a descending order.

/* Evaluating the weight of each link by formula (4).*/

Let $E_{\text{tmp}} = E$.

dowhile ($E_{\text{tmp}} \neq \emptyset$)

 choose a link, say $e_{i,j}$, from E_{tmp} .

$w(e_{i,j}) = p_{i,j}$.

 Let $k = 1$.

dowhile ($k \leq n$)

 if (link $e_{i,k}$ and $e_{k,j}$ are exist in E)

$w(e_{i,j}) = w(e_{i,j}) + (1 - w(e_{i,j})) * p_{i,k} * p_{k,j}$.

 end if

$k = k + 1$.

end dowhile

Let $E_{\text{tmp}} = E_{\text{tmp}} - \{e_{i,j}\}$ /* discard $e_{i,j}$ from E_{tmp} */

end dowhile

Sorting all links according to their weight in a descending order.

step 2 Choosing the first two weightiest nodes as starting node, say v_i and v_j , for obtaining an adequate k -node set (say G_k). Note that G_k is $\{v_i, v_j\}$.

dowhile($c(G_k) < C_{\text{constraint}}$)

 Find v_i , such that $w(v_i) = \max\{w(v_i) | v_i \in G, v_i \notin G_k\}$.

 Let $G_k = G_k \cup \{v_i\}$.

 Let $c(G_k) = c(G_k) + c(v_i)$.

end dowhile

step 3 Choosing the weightiest link, say $e_{i,j}$. Selecting v_r, v_j which are incident with $e_{i,j}$ as starting nodes for obtaining another adequate k -node set (say G'_k). Note that G'_k is $\{v_r, v_j\}$.

dowhile($c(G'_k) < C_{\text{constraint}}$)

 Find $e_{i,j}$, such that $w(e_{i,j}) = \max\{w(e_{i,j}) | v_i \in G'_k, v_j \in V_{\text{adj}}(G'_k)\}$.

 Let $G'_k = G'_k \cup \{v_j\}$.

 Let $c(G'_k) = c(G'_k) + c(v_j)$.

 end dowhile

step 4 /*Performing reduction processing*/

/*if a subset of the k -node set satisfied capacity constraint, the k -node set is replaced by its subset.*/

Find v_i , such that $c(v_i) = \min\{c(v_i) | v_i \in G_k\}$.

dowhile($(c(G_k) - C_{\text{constraint}}) > c(v_i)$ and $|k| > 2$)

 Let $G_k = G_k - \{v_i\}$ /* discard v_i from G_k */

 Let $c(G_k) = c(G_k) - c(v_i)$.

 Find v_i , such that $c(v_i) = \min\{c(v_i) | v_i \in G_k\}$.

end dowhile

Find v_i , such that $c(v_i) = \min\{c(v_i) | v_i \in G'_k\}$.

dowhile($(c(G'_k) - C_{\text{constraint}}) > c(v_i)$ and $|k| > 2$)

 Let $G'_k = G'_k - \{v_i\}$ /* discard v_i from G'_k */

 Let $c(G'_k) = c(G'_k) - c(v_i)$.

 Find v_i , such that $c(v_i) = \min\{c(v_i) | v_i \in G'_k\}$.

end dowhile

step 5 /*Computing the reliability of the two k -node sets and outputting the optimal k -node set*/

if ($G_k = G'_k$) /*the two selected k -node sets are same*/

 compute $R(G_k)$.

 output $G_k, R(G_k)$.

else /*the two selected k -node sets are different*/

 compute $R(G_k), R(G'_k)$.

 if ($R(G_k) > R(G'_k)$)

 output $G_k, R(G_k)$.

 else

Table 1
Comparison with other methods

Size		Global optimal solution		Exhaustive ^a		EM ^b		KM ^c		Proposed method	
n^d	e^e	Max_rel ^f	k -node set ^g	NRC ^h	err ⁱ	NRC ^h	err ⁱ	NRC ^h	err ⁱ	NRC ^h	err ⁱ
5	6	0.9462500	1,2,3	32	0	10	0	9	0.0122881	2	0
6	8	0.9383065	4,5,6	64	0	15	0	12	0.0444220	2	0
6	9	0.9950069	1,3,5	64	0	20	0	12	0.0280687	1	0
7	8	0.9187206	1,2,4	128	0	35	0	15	0	1	0
7	11	0.9967785	1,2,3	128	0	35	0	15	0.0200188	1	0
8	10	0.9793787	1,6,8	256	0	45	0	18	0.0289410	2	0
8	11	0.9974378	4,6	256	0	44	0	13	0.0361958	1	0
10	13	0.9347952	1,7,8,9,10	1024	0	255	0	35	0.2155169	2	0
10	17	0.9994068	2,8,9	1024	0	119	0	24	0.0074441	2	0
10	19	0.9995282	1,5,6	1024	0	150	0	24	0.0019527	2	0
11	17	0.9974023	1,10,11	2048	0	135	0	27	0.0120129	2	0
12	18	0.9858263	3,4,5,6	4096	0	538	0	38	0.0299250	2	0
12	21	0.9990777	1,3,5,6	4096	0	537	0	38	0.0056617	2	0.0006069
13	20	0.9978402	4,6	8192	0	246	0	23	0.0189070	2	0
19	31	0.9979870	6,8,9	524288	0	2369	0	51	0.0856661	1	0
30	30	0.9856600	4,5,6	2 ³⁰	0	58325	0	84	0.0706510	1	0
32	33	0.9937695	2,3,4	2 ³²	0	163310	0	90	0.1050981	1	0
60	60	0.9530911 ^j	5,6	2 ^{60j}	– ^k	– ^k	– ^k	117	0.1408611	1	0
120	120	0.9800450 ^j	2,3	2 ^{120j}	– ^k	– ^k	– ^k	237	0.1812755	1	0
240	240	0.9809699 ^j	4,5	2 ^{240j}	– ^k	– ^k	– ^k	577	0.2596127	1	0

^a Exhaustive: the exhaustive method.

^b EM: the exact method [11].

^c KM: the k -tree reduction method [12].

^d n : the number of nodes in G , $n = |V|$.

^e e : the number of links in G , $e = |E|$.

^f Max_Rel: maximum reliability satisfies our constraint (namely, exact solution.)

^g k -node set: the nodes we selected.

^h NRC: the number of reliability computation.

ⁱ err: the absolute error from exact solution (namely, Max_Rel).

^j The value is obtained by manual calculation.

^k –: denotes that the value is barely obtained.

```

output  $G'_k, R(G'_k)$ .
end if
end if
end KNR

```

3.3. An illustrative example

Fig. 1 illustrates the topology of a DCS with eight nodes and eleven links. The problem involves determining a subset of the DCS which includes some of nodes v_1, v_2, \dots, v_8 whose total capacity at least as large as the capacity constraint of one hundred units.

In step 1, each node's weight is evaluated by formula (2). The weights of v_1, v_2, \dots, v_8 are 0.998537, 0.9835, 0.9865, 0.9998898, 0.9766, 0.9995756, 0.9696 and 0.999664, respectively. Herein, all nodes are sorted according to their weights in a descending order. Thus, each node's weight is $v_4, v_8, v_6, v_1, v_3, v_2, v_5, v_7$ in order. According to formula (4), we obtain the weight of $e_{1,2}, e_{1,7}, \dots, e_{6,8}$ which are 0.89, 0.81, 0.93, 0.85, 0.91, 0.921094, 0.9441, 0.989216, 0.962482, 0.84, 0.975616, respectively. These links are sorted according to weight. The outcome is

$e_{4,8}, e_{6,8}, e_{5,6}, e_{4,6}, e_{1,8}, e_{4,5}, e_{3,4}, e_{1,2}, e_{2,3}, e_{6,7}, e_{1,7}$ in a descending order.

In step 2, nodes are selected from the sorted list of nodes to generate the first k -node set. Initially, v_4 and v_8 are selected. Notably, the sum of their capacities is 94 and does not satisfy our capacity constraint. Next, the node v_6 is included. The sum of capacity of these three nodes is 143, which satisfies the capacity constraint. By doing so, the first K -node set $\{v_4, v_8, v_6\}$ is obtained.

In step 3, nodes are selected according to the sorted list of edges to generate the second k -node set. Initially, v_4 and v_8 which are incident with the weightiest link (say $e_{4,8}$) are selected. Notably, the sum of capacity is less than capacity constraint. Next, we check the other link (e.g. $e_{3,4}, e_{4,5}, e_{4,6}, e_{1,8}$ and $e_{6,8}$) which are adjacent to nodes v_4 or v_8 . Where $e_{6,8}$ denotes the weightiest link among those links. Therefore, we select v_6 which is incident with $e_{6,8}$. We get the second k -node set $\{v_4, v_8, v_6\}$ which is same as the first k -node set.

In step 4, if the capacity constraints subtracted from the sum of capacities of selected nodes is greater than or equal to some node, that node is discarded. Repeat this step until no more node can be canceled. Owing to that the capacity of

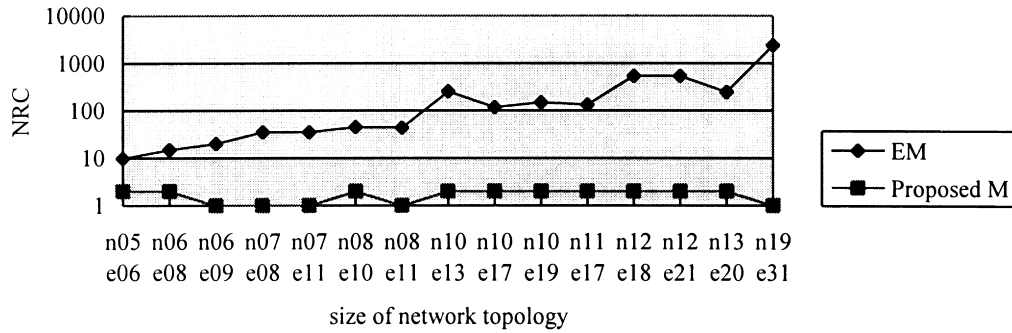


Fig. 2. The number of reliability computation (NRC) of the EM and the proposed method.

node v_8 is 41, we cancel the node v_8 from the k -node set. Consequently, the k -node set becomes $\{v_4, v_6\}$ which still satisfies the capacity constraints.

In step 5, the reliability of the k -node set $\{v_4, v_6\}$ is computed. The result is the same as the k -node set, which is derived by an exhaustive method. Thus, the k -node set $\{v_4, v_6\}$ is obtained, which has the maximum reliability. The number of reliability computation is one.

3.4. Simulation

The accuracy and efficiency of the proposed algorithm are verified by implementing simulation program C language that are executed on a Pentium 100 with 16M-DRAM on MS-Windows 95. We use many network topologies and generate several hundreds of data for simulation. The reliability of each link and the capacity of each node are generated by random number generator. For verifying the sensitivity of our proposed algorithm, the reliability of each link is given in some different ranges. When the number of nodes in topology is very large, such as exceed 60, obtaining the global optimal solution by the exhaustive method or the EM is nearly impossible. For the reason of verifying the correctness of proposed algorithm, when the topology is very large, we constitute the topology in low connectivity.

4. Results and discussion

Results obtained from our algorithm are compared with those of the exhaustive method, the EM and the k -tree reduction method. Table 1 summarizes those results. Fig. 2 demonstrates the number of reliability computation of the EM and our proposed algorithm(only part). Fig. 3 displays the absolute error from exact solution of the k -tree reduction method and our proposed algorithm (only part). Although the exhaustive method and the EM can obtain a global optimal solution, the number of reliability computations is increased exponentially according to the number of processing elements of a DCS. The k -tree reduction method has improved the number of reliability computations, which is equal to $kn - (k + 1)k/2$ where n denotes the number of nodes in a DCS and k represents the number of nodes of the k -node set. However, the absolute error is not very good. The number of reliability computations for the proposed algorithm is constant, which is independent of the topology of DCS. Therefore, the proposed method saves more execution time than the EM and the k -tree reduction method for a large DCS. In addition, the absolute error is smaller than those obtained by the k -tree reduction method.

Although capable of providing the optimal solution, conventional techniques such as the exhaustive method and the EM cannot effectively reduce the number of reliability computations. An application occasionally requires an

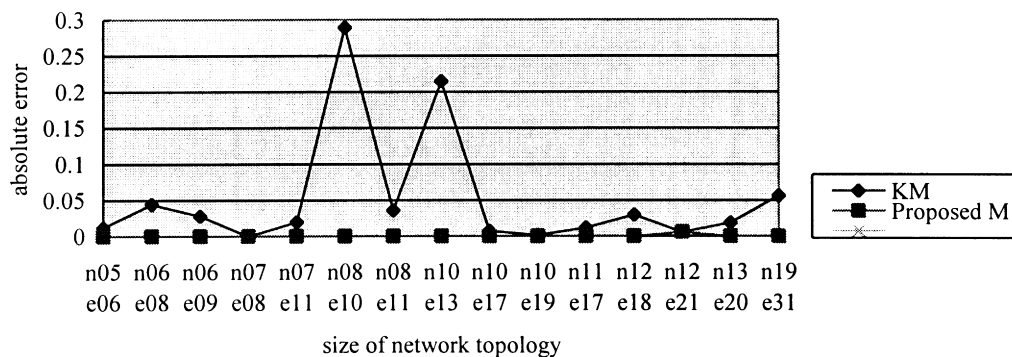


Fig. 3. The absolute error (err) of the k -tree reduction method and the proposed method.

effective algorithm of computing reliability owing to resource considerations. In these circumstances, deriving the optimal reliability may not be a promising option. Instead, an effective algorithm providing approximate reliability is preferred. The k -tree reduction method can reduce computational time in a moderate DCS, but the deviation from an exact solution is not very precise.

In contrast to the computer reliability problem, which is static-oriented, the KNR problems in the DCS are dynamic-oriented since many factors such as node capacity, DCS topology, link reliability, and the number of paths between each node can significantly affect the efficiency of the algorithm [6,13,20,21]. Thus, quantifying the time complexity exactly is extremely difficult. The complexity of the EM is $O(2^e 2^n)$, where e denotes the number of links and n represents the number of processing elements. The complexity of the k -tree reduction method is $O(2^e n^2)$. In our proposed algorithm, in the worst case, the complexity of evaluating the weight of each node is $O(e)$, sorting nodes processing is $O(n \log n)$, evaluating the weight of each link is $O(en)$, sorting links processing is $O(e \log e)$, in the worst case, where topology is complete one, e is equal to $n(n-1)/2$, therefore, $\log e$ is smaller than n , selecting the first k -node set is $O(n)$, selecting the second k -node set is $O(e)$, and computing the reliability of a k -node set using SYREL is $O(m^2)$, where m denotes the number of paths of a selected k -node set [19]. Therefore, the complexity of the proposed algorithm is $O(en + m^2)$. In the k -tree reduction method, which obtains the exact solution below 10%, the average deviation from exact solution exceeds 6.5%. In our simulation case, the number of reliability computation of the proposed algorithm is constant. We generate testing data, namely the reliability of each link, according to the range of $0.0 \sim 1.0$, $0.5 \sim 1.0$, $0.8 \sim 1.0$ and $0.95 \sim 1.0$. Consequently, we obtained information as follows. The exact solution can be obtained about 80, 80, 90, and 80%, respectively. The average deviation from exact solution is under 0.03, 0.02, 0.004 and 0.0002, respectively. The error bound is under 0.14, 0.06, 0.04 and 0.001, respectively. In some cases, the number of paths (in which the length is between 3 and $n-1$) of another set of nodes is much more than those of selected k -node set. Notably, the proposed algorithm cannot obtain the exact solution.

5. Conclusions

This work presents a novel heuristic algorithm to derive a k -node set with capacity constraint of maximal reliability. The proposed algorithm is compared with the EM and the k -tree reduction method for various topologies. According to that comparison, the proposed algorithm is more efficient in execution time for a large DCS than those methods. The proposed algorithm based on taking a short time to evaluate the weight of each node and the weight of each link. According to the weights of nodes and links, the algorithm

can accurately predict which node will be included to obtain a better k -node set. The proposed algorithm can also effectively reduce the number of reliability computations. Because the number of reliability computations is either one or two, the proposed algorithm can provide the desired performance. Further, when the proposed algorithm fails to provide an exact solution, the deviation from the exact solution is only slight.

References

- [1] A.S. Tanenbaum, Computer Networks, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [2] W. Stallings, Local Networks, Macmillan, New York, 1984.
- [3] Anup Kumar, Dharm P. Agrawal, A generalized algorithm for evaluation distributed program reliability, IEEE Trans. Reliab. 42 (3) (1993) 416–426.
- [4] J.A. Stankovic, A perspective on distributed computer systems, IEEE Trans. Computers 33 (1984) 1102–1115.
- [5] A. Satyanarayana, J.N. Hagstrom, New algorithm for reliability analysis of multiterminal networks, IEEE Trans. Reliab. R-30 (1981) 325–333.
- [6] M.A. Aziz, Pathset enumeration of directed graphs by the set theoretic method, Microelectron. Reliab. 37 (5) (1997) 809–814.
- [7] S. Rai, A.K. Sarje, E.V. Prasad, A. Kumar, Two recursive for computing the reliability of k -out- n systems, IEEE Trans. Reliab. 36 (2) (1987) 261–265.
- [8] K.B. Irani, N.G. Khabbaz, A methodology for the design of communication networks and the distribution of data in distributed super-computer systems, IEEE Trans. Computers C-31 (1982) 420–434.
- [9] L. Kleinrock, Analytic and simulation methods in computer network design, Proc. Spring Joint Computer Conf. (1970) 569–579.
- [10] H. Frank, W. Chou, Topological optimization of computer networks, Proc. IEEE 60 (1972) 1385–1397.
- [11] R.S. Wilkov, Analysis and design of reliable computer networks, IEEE Trans. Commun. COM-20 (1970) 660–678.
- [12] D. Torrieri, Calculation of node-pair reliability in large networks with unreliable nodes, IEEE Trans. Reliab. 43 (3) (1994) 375–377.
- [13] J.S. Provan, Bounds on the reliability of networks, IEEE Trans. Reliab. 16 (1986) 319–329.
- [14] D.W. Coit, A.E. Smith, Reliability optimization of series-parallel systems using a genetic algorithm, IEEE Trans. Reliab. 45 (2) (1996) 254–260.
- [15] C.S. Raghavendra, S. Hariri, Reliability optimization in design of distributed systems, IEEE Trans. Software Engng SE-11 (10) (1985) 1184–1193.
- [16] K.K. Aggarwal, Y.C. Chopra, J.S. Bajwa, Topological layout of links for optimizing the S-T reliability in a computer communication system, Microelectron. Reliab. 22 (1982) 341–345.
- [17] Ruey-Shun Chen, D.J. Chen, Y.S. Yeh, Reliability optimization of distributed computing systems subject to capacity constraint, J. Computers Math. Appl. 29 (4) (1995) 93–99.
- [18] Ruey-Shun Chen, D.J. Chen, Y.S. Yeh, A new heuristic approach for reliability optimization of distributed computing systems subject to capacity constraints, J. Computers Math. Appl. 29 (3) (1995) 37–47.
- [19] Salim Hariri, C.S. Raghavendra, SYREL: a symbolic reliability algorithm based on path and cutset methods, IEEE Trans. Computers C-36 (10) (1987) 1224–1232.
- [20] Wesley W. Chu, Optimal file allocation in a multiple computer system, IEEE Trans. Computers C-18 (1969) 10.
- [21] M.S. Lin, D.J. Chen, New reliability evaluation algorithms for distributed computing systems, J. Inform. Sci. Engng 8 (3) (1992).

Chin-Ching Chiu—Education: He is currently working towards the PhD degree in Department of CS & IE, National Chiao-Tung University. Sept. 1988–June 1991 MS in Computer Science, Department of EE & IE, Tamkang University. He received a BS degree in Computer Science from Soochow University. Professional Background: Sept. 1991 till present Instructor, Department of Information Management, Tak-Mi Colloge. April 1984–Aug. 1991 Computer Engineer in Data Communication Institute of Directorate General of Telecommunications. Oct. 1982–April 1984 System Analyst of SYSCOM computer company. Research Interest: Reliability Analysis of Network and Distributed System.

Ruey-Shun Chen—Education: Sept. 1991–April 1995 PhD in Department of CS & IE, National Chiao-Tung University. Professional Background: Aug. 1995 till present Associate Professor, Institute of Information Management, National Chiao-Tung University. Research Interest: Reliability and Performance Evaluation: Networks, Interconnection Networks, and Distributed systems.

Yi-Shiung Yeh—Education: Sept. 1981–Dec. 1985 PhD in Computer Science, Department of EE & CS, University Of Wisconsin-Milwaukee. Sept. 1978–June 1980 MS in Computer Science, Department of EE & CS, University Of Wisconsin-Milwaukee. Professional Background: Aug. 1988 till present Associate Professor, Institute of CS & IE, National Chiao-Tung University. Jul. 1986–Aug. 1988 Assistant Professor, Department of Computer & Information Science, Fordham University. Jul. 1984–Dec 1984 Doctorate Intern, Johnson Controls, Inc. Aug. 1980–Oct. 1981 System Programmer, System Support Div., Milwaukee County Gov. Research Interest: Data security & Privacy, Information & Coding Theory, Game Theory, Reliability and Performance.