Journal of Telecommunications System & Management

Petingi et al., J Telecommun Syst Manage 2018, 7:1
DOI: 10.4172/2167-0919.1000163

Research Article

Open Access

Applying Parallelism to Calculate the Number of Path-Sets and Cut-Sets of a Graph with Restricted Diameter Based on Construction-Destruction Methods

Louis Petingi[1]*, Ilya B Gertsbakh[2] and Steven Wojcik[1]

[1]Department of Computer Science, CUNY College of Staten Island, Staten Island, New York 10314, USA
[2]Department of Mathematics and Computer Science, Ben Gurion University of the Negev, Beer-Sheva 8410501, Israel

Abstract

Consider a probabilistic graph G comprising a node-set V, an edge-set E, and where $K \subseteq V$ is a set of nodes called the terminal nodes. The edges of the network fail independently with known probabilities and nodes are perfectly reliable. We present a Monte Carlo Construction technique to evaluate the number of sub-graphs of G, called the d-path-sets of G, in which the distance between terminals is less or equal a given integer value d. Calculation of the number of d-path-sets permit us to evaluate the Diameter Constrained Reliability of a network, an extension of the classical reliability, both known to be intractable problems; because of this intractability these Monte Carlo methods are applied and embedded within a distributed/parallel environment, using MPI libraries. Construction-Destruction methods in combination with parallel computing, provide accurate estimates.

Keywords: Construction-Destruction MC methods; Path- and Cut - sets; Diameter-constrained network reliability; Parallelism

Introduction

Let $G=(V,E,K)$ be a probabilistic graph where V is a finite set called the node-set (or vertices) and E is a finite set called the edge-set (or links), and $K \subseteq V$ is called the terminal set. In this probabilistic reliability model edges fail (survive) independently with pre-defined probabilities, and vertices never fail. The classical reliability (CR), $R_k(G)$, is the probability for each pair of terminals u and v, a u,v- path exists. Given an integer d, the Diameter Constrained Reliability, $R_k(G,d)$, DCR for short, introduced in 2001 [1], gives the probability that there exist an operational path composed of d edges or less, for every terminal pair u and v, or, equivalently, the maximum distance (i.e., K - diameter) between pairs $u, v \in K$ is d or less. If some nodes of K belong to different connected components then the K-diameter=∞.

The DCR (see [2,3]) can measure, for example, the performance of communication networks based upon the distribution of packets with a time-to-live constraint on the maximum number of hops (nodes) they can traverse (i.e., IPv6 packets [4]).

The DCR subsumes the classical reliability measure; as the maximum number of edges in path of a graph with n nodes is n-1, then $R_k(G,d) = R_k(G)$, for the case d=n-1. Computation of the classical reliability, for arbitrary terminal set K, is NP-hard [5], so it is then calculation of the DCR. Construction-Destruction Methods (CDM) [6, 7, 8, 9] were originally introduced to estimate the classical reliability. This work is primarily concerned with the study of the Construction Monte Carlo technique (CM for short) to estimate the number of topological structures called d-path-sets; this allows as a by-product, to accurately evaluate the DCR.

A path-set of a network G is a set of j edges that spans a sub-graph in which the set terminal nodes are contained within a component; let S_j represent the number of path-sets with j edges. A set of j links is a cut- set if its removal from a network results in a sub-graph in which a pair of nodes of K will be on different connected components. The notation C_j is used to represent the number of cut-sets with j edges. *CDM* yielded excellent estimates for both S_j and C_j, and therefore, as a by-product, for the classical reliability.

The notion of path-set can be extended when an integer d is given, that is, a d-path-set is a sub-graph in which all terminal vertices belong to a single connected component with K-diameter ≤ d. A d-cut-set is a set of edges that if removed from a graph will yield a sub-graph with K-diameter > d. In this manuscript, CM is particularly tailored to find the number of d-path-sets and d-cut-sets with a given number of edges, consequently allowing to precisely estimate the DCR. The relevance of this contribution is that depending on the terminal set K, computation of the number of cut-sets and path-sets belongs to the #P-Complete computational class [10].

Construction Method to Estimate the Number of Path- and Cut-sets of Specific Sizes and Evaluation of the DCR

Construction model

For an edge x, $q(x)$=1-$p(x)$ is the failure probability of x, and $p(x)$ is the probability of survival of x.

Let d be an integer and under the assumption that edges (links) fail equally with probability $q = 1-p$, the Diameter Constrained Reliability, for a graph with $m = |E|$ edges, can be computed as [11]:

$$R_K(G,d,p) = \sum_{j=l_d}^{m} S_j^d p^j (1-p)^{m-j}, \tag{1}$$

where the cardinality of the set of d-path-sets composed of exactly j edges is denoted as S_j^d and where l_d is the cardinality of a minimum-path-set (d-*min -path*). As it was aforementioned, because of the intractability of evaluating the DCR, computation of one or more S_j^d

***Corresponding author:** Louis Petingi, Department of Computer Science CUNY College of Staten Island, Staten Island, New York 10314, USA, Tel: +17189822844, E-mail: louis.petingi@csi.cuny.edu

Received June 25, 2018; **Accepted** July 03, 2018; **Published** July 07, 2018

Citation: Petingi L, Gertsbakh IB, Wojcik S (2018) Applying Parallelism to Calculate the Number of Path-Sets and Cut-Sets of a Graph with Restricted Diameter Based on Construction-Destruction Methods. J Telecommun Syst Manage 7: 163. doi: 10.4172/2167-0919.1000163

belongs to the #P-Complete computational class. The Construction technique is as follows: a random permutation of the edges $\Pi=\{x_1,x_2,...,x_m\}$ is generated, and all the edges are set *DOWN*; then turn *UP* an edge *x*, one at the time, left to right, until the sub-graph generated becomes *UP*; if the DOWN →UP transition happens at step *s*, we then say that the *critical-number* of the Π is *s*.

Within the context of the DCR, *UP* is the state (sub-graph) in which a path of length $L \le d$ exist for every pair of terminals in *K*. The condition that the state is *UP* or not could be verified by Floyd 's procedure [12] of complexity $O(n^3)$ for a network with *n* vertices.

Let g_s corresponds to the transition probability of the event that the system goes from DOWN→UP at step *s* of the CM. Let $g=\{g_1,...,g_m\}$ be a collection of probabilities called the *C-spectrum* [8]. Let the *Cumulative*-C-spectrum the collection of probabilities $y=\{y(1),...,y(m)\}$ in which $y(k)=\sum_{s=1}^{k} g_s$. As S^d_k is the number of *d*-path-sets with exactly *k* edges, that is, having *k* edges *up* and remaining *m-k down*, then

$$S_k^d = y(k)\frac{m!}{k!(m-k)!}. \tag{2}$$

In the CM, *M* random permutations (trials) are simulated and let N_i count the number of them with the DOWN→UP transition at step *i*-th of the C-process; then $\sum_{j=1}^{k} N_j / M$ is an unbiased estimate of of $y(k)$, under the assumption that all Π's are equi-probable. For the proof of CM see [8], in which a proof is stated for the dual variation of the Construction MC method, i.e., the Destruction MC method, also under the frame of the classical reliability.

Let $y(k)$=*Probability* {from a *DOWN*-state → *UP*-state happens before or at the *k* step of the process}. The Relative-Error for the Unbiased-Estimator of $y(k)$ is

Remark 1: $re[\widehat{y(k)}] = \sqrt{1-\widehat{y(k)}} / \sqrt{M \cdot \widehat{y(k)}}$. *In addition, for fixed integer M (i.e., number of trials),* $re[\widehat{y(k)}]$ *is monotonically-non-increasing as K increases.*

Let *C* be a set of edges of a graph *G* and *j* be an integer; then

$C_j^d = \#\{C : |C| = j$ *and the sub-graph resulting by removing C from G has K*-diameter > *d* }.

The following function relates *d*-path-sets with *d*-cut-sets [11]

$$C_j^d + S_{m-j}^d = \binom{m}{j}. \tag{3}$$

Thus calculating C^d_j is equivalent to evaluate S^d_{m-j} and vice versa.

Construction algorithm

The following algorithm appears [11]:

1. Input: Graph $G=(V,E,K)$, $K \subseteq V$, and an integer *d*.

2. $N_j = 0, 1 \le j \le m$ (N_j - numb. of permts. with critical-number *j*).

3. **Repeat** *M* times, where *M* is the number of trials:

(a) Randomly generate.

$\Pi=\{x_1,x_2,...,x_m\},\{x_i \in E\}$.

(b) Initiating with the graph with no edges, include edges from permt. Π, left to right, and stop when the *K*-diameter $\le d$ (i.e., Floyd's).

(c) If the first edge UP is x_s then *s* is the critical-number of the permt. Π;

$N_s = N_s + 1$

4. **End Repeat**.

5. The C-Spectrum is $g=\{g_1,g_2,...,g_m\}$, where $g_s = \frac{N_s}{M}$.

6. The Cumulative-C-Spectrum defined on $y=\{y_1,y_2,...,y_m\}$, where

$$y_s = \sum_{j=1}^{s} g_j.$$

7. $S_r^d = y_r \frac{m!}{r!(m-r)!}, 1 \le r \le m.$

Parallization Using MPI

In this section we embed the CM algorithm described in Section 2.2, within a parallel-distributed environment, using MPI [13], Message-Passing-Interface programming libraries:

1. Input: Graph $G=(V, E, K)$, $K \subseteq V$, and an integer *d*.

2. Let $p=\{p_1,p_2,...,p_z\}$ be the collect. of *z* processors and M_j be the number of trials to be generated for processor p_j.

3. For processors $p_1,p_2,...,p_Z$, let $N_k^j = 0, 1 \le j \le z; 1 \le k \le m$ (N_s^j is # of permts. with critical-number *s*, evaluated by processor p_j):

(a) For processor p_j, $1 \le j \le z$, **repeat** M_j times:

i) Randomly generate

$\Pi=\{x_1,x_2,...,x_m\},\{x_i \in E\}$.

ii) Include edges from Π, left to right until the *K*-diameter ≤ *d*.

iii) If x_s is the first edge UP, then $N_s^j = N_s^j + 1$.

(b) **End repeat**.

(c) Send # of critical hits, $N_s^j, 1 \le j \le z, 1 \le s \le m$, to the master processor p_1 (using MPI-reduce [13]).

4. The master processor p_1 generates the C-Spectrum, the Cumulative C-Spectrum, and the number of *d*-path-sets with specific number of edges.

(a) Let the total trials be defined as $M = \sum_{j=1}^{z} M_j$.

(b) Let the total # of permts. with critical-number *i* be defined as

$N_i = \sum_{j=1}^{z} N_i^j, 1 \le i \le m$.

(c) Let the C-Spectrum be defined as the collection of $g=\{g_1,g_2,...,g_m\}$ g_s=Prob {the critical-number=*s*}, meaning $g_s = N_s/M$.

(d) Let the Cumulative C-Spectrum. $y=\{y_1,y_2,...,y_m\}$ where y_s= Prob {critical-number is ≤ *s*}, i.e.,

$$y_s = \sum_{i=1}^{s} g_i, 1 \le s \le m.$$

(e) $S_r^d = y_r \frac{m!}{r!(m-r)!}, 1 \le r \le m.$

Computational Tests

The parallelization of CM (PCM) discussed in the previous section, was coded in C++ and using MPI libraries. The tests were run on Salk, a cluster with 1,028 processor cores (2.3 GHz), each with 2 gigabytes of memory. To test a graph using PCM takes $M.\log_2 m.n^3$ steps in which *n* and *m* are the number of nodes and edges of the graph, respectively, and *M* is the number of trial (permutations) generated.

The 4-dimensional-hypercube (4DH) and Dodecahedron (Dode) (see Figure 1) graphs were tested. For the 4DH we let $K=\{s,t\}$, were *s* and *t* were at a distance 4. Meanwhile for the Dode we let $K=\{0, 19\}$ (Table 1).

The numbers depicted in Table 2 are tests conducted on the Dode graph for d either 7 or 15, meanwhile Table 3 illustrates the tests conducted on the 4DH also for the same values of d.

Column 1 of Tables 2 and 3 indicates the diameter bound d applied for the tests performed and column 2 describes the number of processors used for each test. Rows 2 and 9 of Tables 2 and 3 shows the number of d-path-sets with specific number of edges calculated by an algorithm using backtracking of non-polynomial time-complexity. Rows 3 and 4, 5 and 6, 7 and 8, 10 and 11, 12 and 13, and 14 and 15 show the estimates yielded by the PCM and the error with respect to the exact values, for 1, 64 and 128 processors. The results show a reduction in the percentage error of the estimates when more processors were concurrently run (or equivalently more trials/permutations were simulated), with a maximum error of 0.6 and 0.006 (on 128 processors), when both the Dode and 4DH were tested. From eqn. (1) the DCR can be also estimated in terms of the d-path-sets (Table 1).

Graphical representations of Tables 2 and 3 are illustrated in Figures 2 and 3 respectively, when d=7. S_r^d, $1 \le r \le m$, are accurately estimated whenever $r \to m$, and more processors (trials/permutations) were applied in the analysis. This observation follows from the fact that the Relative-Error of $y(r)$ (i.e., S_r^d) is monotonically-non-increasing (see Remark 1, Section 2.1).

The results suggest useful applications of CM in combination with parallel programming (i.e., PCM). For example, calculating the size of a d-min-cut of a graph can be determined by finding the first integer for which $S_{m-j}^d \neq \binom{m}{j}$ (see eqn. (3)), thus yielding the number C_j^d of such cuts for the 4-dimensional Hypercube (Table 3) and when d=7 or d= 15 then j=4, m-j=28. For K={s,t} and whenever d is not a fixed value, size of a d-min-cut can be obtained by the Max-Flow-Min-cut Theorem [14]; however it is an intractable problem for paths of bounded-length d, for fixed $d > 4$ [15].

In the next section we concentrate on determining the accuracy of estimating the size of d-min-cuts, for a family of graphs in which these values are known, so tests on larger graphs can be conducted.

Size of Min-Cuts

This section is advocated to measure the accuracy of PCM when calculating the cardinality of a d-min-cut of a network.
As explained in the previous section, whenever $d > 4$ [15], and K={s, t}, the problem of finding a min-cut is an intractable one. For these tests, we introduce a new family of graphs, SW ($mult$, D), with two terminal vertices K={s,t}, and two parameters, $mult$, and D, where

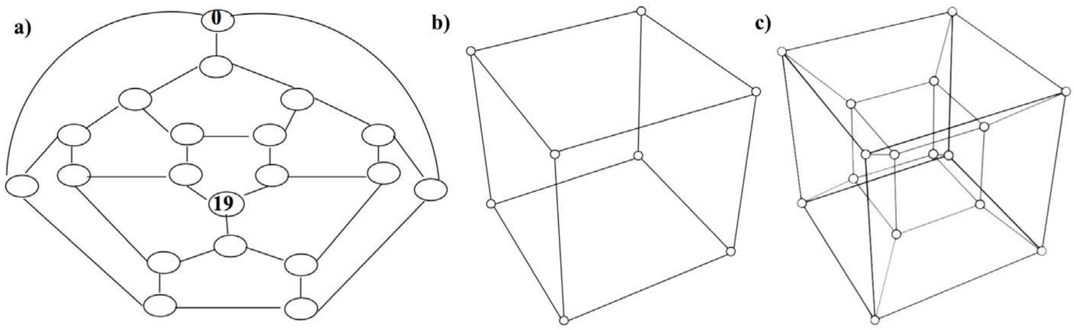

Figure 1: Types of topologies: a) Dodecahedron, b) 3-dimensional and c) 4-dimensional Hypercubes.

Topo	K=(s,t)	Exact-Reliab.	np	Estim-Reliab.	% error
Dodecahedron	(0,19)	0.9967389962	1	0.99673967938	6.85E-05
			64	0.99674238793	3.40E-04
			128	0.99674246075	3.48E-04
4-dim Hypercube	(0,10)	0.9997922232	1	0.99979041258	1.81E-04
			64	0.99979171835	5.05E-05
			128	0.99979251006	2.87E-05

Table 1: Estimated DCR for Dode and the 4DH topologies, under the assumption that d=7, unique edge reliability 0.9 and M=10^6 trials per processor were generated.

d			1-4	5	6	7	8	9	10	15	18	22	25	26	27	28	29	30
		exact	0	6	162	2094	17226	101160	450957	32494118	46188065	5423592	141618	27345	4058	435	30	1
	1 Proc.	estimated	0	3	150	2038	17412	102625	455032	32486572	46186863	5422460	141627	27345	4058	435	30	1
		% error	0	50	7.40741	2.67431	1.07976	1.44820	0.90363	0.02322	0.00260	0.02087	0.00636	0	0	0	0	0
7	64 Proc.	estimated	0	6	163	2107	17287	101459	451900	32495807	46194427	5423801	141620	27345	4058	435	30	1
		% error	0	0	0.61728	0.62082	0.35412	0.29557	0.20911	0.00520	0.01377	0.00385	0.00141	0	0	0	0	0
	128 Proc.	estimated	0	6	161	2094	17198	101068	450575	32491312	46184148	5423572	141620	27345	4058	435	30	1
		% error	0	0	0.61728	0	0.16254	0.09095	0.08471	0.00864	0.00848	0.00037	0.00141	0	0	0	0	0
		exact	0	6	162	2094	17238	101442	454071	34721924	51031196	5531394	141630	27345	4058	435	30	1
	1 Proc.	estimated	0	3	150	2038	17424	102911	458126	34709562	51019672	5530405	141638	27345	4058	435	30	1
		% error	0	50	7.40741	2.67431	1.07901	1.44812	0.89303	0.03560	0.02258	0.01788	0.00565	0	0	0	0	0
15	64 Proc.	estimated	0	6	163	2098	17280	101543	453302	34720684	51032424	5531711	141630	27345	4058	435	30	1
		% error	0	0	0.61728	0.19102	0.24365	0.09956	0.16936	0.00357	0.00241	0.00573	0	0	0	0	0	0
	128 Proc.	estimated	0	6	162	2094	17229	101387	454254	34728135	51032766	5531329	141629	27345	4058	435	30	1
		% error	0	0	0	0	0.05221	0.05422	0.04030	0.01789	0.00308	0.00118	0.00071	0	0	0	0	0

Table 2: Comparing estimates vs. exact values S_i^d as calculated by PCM with the exact number for the Dode topology.

d			1-3	4	5	6	7	8	9	10	16	22	25	26	27	28	29	30	31	32
		exact	0	24	672	9132	79968	505062	2439504	9324616	465201161	63642792	3359104	905428	201320	35958	4960	496	32	1
	1 Proc.	estimated	0	25	682	9154	80040	506730	2440049	9315438	465379880	63633519	3358973	905422	201319	35958	4960	496	32	1
		% error	0	4.1667	1.48810	0.24091	0.09004	0.33026	0.02234	0.09843	0.03842	0.01457	0.00390	0.00066	0.00050	0	0	0	0	0
7	64 Proc.	estimated	0	24	671	9136	79948	505032	2439436	9324825	465165003	63643445	3359095	905424	201320	35958	4960	496	32	1
		% error	0	0	0.14881	0.04380	0.02501	0.00594	0.00279	0.00224	0.00777	0.00103	0.00027	0.00044	0	0	0	0	0	0
	128 Proc.	estimated	0	24	672	9135	79965	505218	2439360	9324030	465181655	63642366	3359112	905428	201320	35958	4960	496	32	1
		% error	0	0	0	0.03285	0.00375	0.03089	0.00590	0.00628	0.00419	0.00067	0.00024	0	0	0	0	0	0	0
		exact	0	24	672	9132	79968	505518	2448912	9415732	480025073	63644808	3359104	905428	201320	35958	4960	496	32	1
	1 Proc.	estimated	0	25	682	9154	80040	507182	2449586	9406078	480402682	63635712	3358973	905422	201319	35958	4960	496	32	1
		% error	0	4.1667	1.48810	0.24091	0.09004	0.32917	0.02752	0.10253	0.07866	0.01429	0.00390	0.00066	0.00050	0	0	0	0	0
15	64 Proc.	estimated	0	24	672	9130	79939	505268	2447419	9410603	480058263	63645131	3359106	905429	201320	35958	4960	496	32	1
		% error	0	0	0	0.02190	0.03626	0.04945	0.06097	0.05447	0.00691	0.00051	0.00006	0.00011	0	0	0	0	0	0
	128 Proc	estimated	0	24	672	9138	79986	505661	2449266	9417088	480027448	63645866	3359118	905429	201320	35958	4960	496	32	1
		% error	0	0	0	0.06570	0.02251	0.02829	0.01446	0.01440	0.00049	0.00166	0.00042	0.00011	0	0	0	0	0	0

Table 3: Comparing estimate vs. exact values S_i^d as calculated by PCM with the exact number for the 4DH topology.

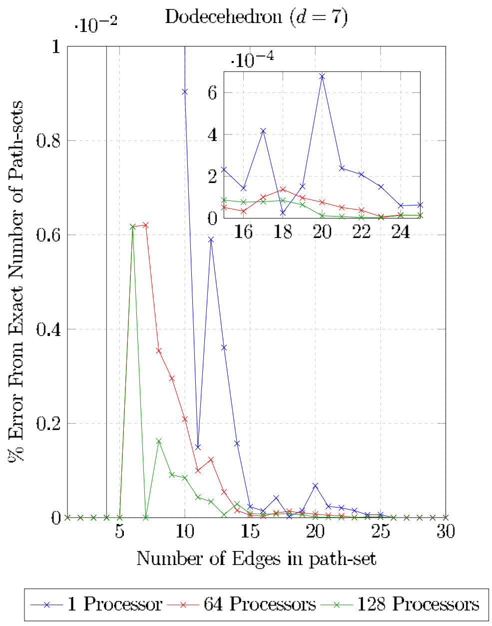

Figure 2: Percentage error of the PCM, comparing estimated with exact values when d = 7, and for 1, 64 and 128 processors, for the Dode graph.

mult is the number of parallel edges connecting any two adjacent vertices of the graph, and *D* is the distance between terminals *s* and *t*. Figure 4 illustrates how a SW(*mult*,*D*) graph is constructed from its parameters. In a SW(*mult, D*) graph, the size of a *d*-min-cut,

s_{min}(SW(*mult, D*)), can be explicitly calculated as function of the parameters, *mult* and *D*, thus avoiding computationally expensive exact evaluation algorithms to determine it.

The following statements are easy to prove:

Claim 1: *For a graph* SW(*mult*,*D*),

1. The number of edges m(SW(*mult*,*d*))=*mult*.(3.D+9)

2. The number of vertices n(SW(*mult*,*D*))=3.*D*+4

3. The size of a *d*-min-cut of SW(*mult*,*D*) is

$$s_{min}(SW(mult,D)) = \begin{cases} 0 : d < D \\ mult : d = D \\ 3 \cdot mult : d > D \end{cases} \qquad (4)$$

As explained previously, s_{min}(SW(*mult*,*D*)) is equal to the first integer j for which $S^d_{m-j} \neq \binom{m}{j}$ (eqn. (3)) or equivalently when *y*(*m*-*j*)<1.00 (eqn. (2)).

Table 4 shows several computational tests conducted on the SW graphs to determine the accuracy to estimate s_{min} (SW (*mult* ,*D*)), known to be an NP-hard problem. Column 1 of this table represents the distance *D* between terminals *s* and *t*, column 2 is the integer *d*, column 3 is the edge multiplicity parameter *mult,* while columns 4 and 5 is the edge-cardinality of the tested graph, and exact cardinality of a *d*-min-cut (eqn. (4)). Let M=10^6 and M= 5 .10^6 (row 2) be the number of trials; row 3 displays the number of processors concurrently run. The rest of columns are the size of *d*-min- cuts (i.e., s_{min}(SW(*mult*,*D*)) as estimated by the PCM. For example consider the third row of the table when *D*=4, *d*=*D*=4, *mult*=3. As a consequence of Claim 1, the number of edges of SW (*mult*,*D*) is *m*=63 and the exact size of a *d*-min-cut is s_{min}(SW(*mult*,*D*))=3 (the column corresponding to the exact size of min-cuts is colored in blue shades). Application of PCM algorithm yields an estimate of the size of the *d*-min-cut

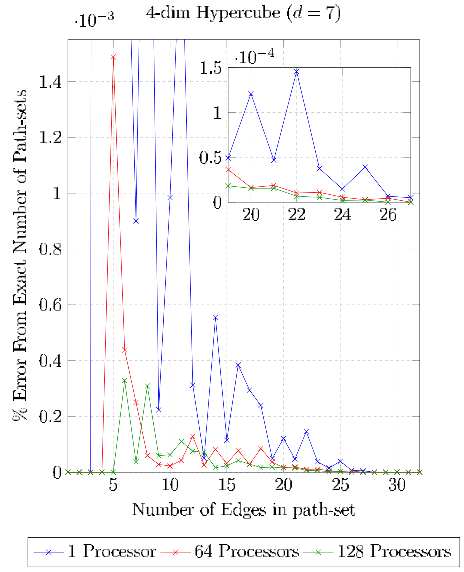

Figure 3: Percentage error of the PCM, comparing estimated with exact values when d = 7, and for 1, 64 and 128 processors, for the 4DH graph.

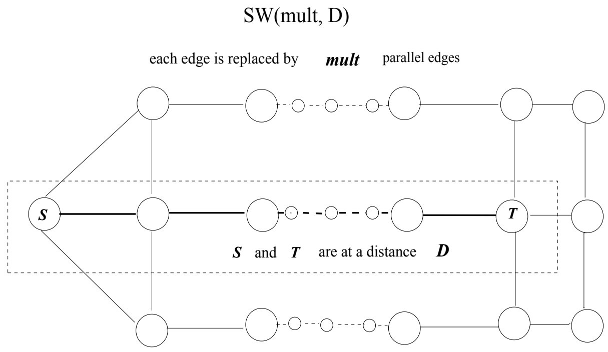

Figure 4: SW(mult,D) family of graphs. The terminal vertices s and t are at a distance D (a unique path of length D connect them). In these graphs each edge is replaced by a bank of mult parallel edges.

										Estimated s_{min}(SW)						
D	**d**	**mult**	**m**	**Exact**			**10⁶ Trials per processor**					**5**	**10⁶ Trials per processor**			
				s_{min}(SW)	1 np	64 np	128 np	256 np	512 np	1024 np	1 np	64 np	128 np	256 np	512 np	1024 np
		3	63	3				3						3		
		4	84	4				4						4		
	4	5	105	5	8			5			6			5		
		6	126	6	13		8	7		6	11				6	
4		7	147	7	21	13	12	10		6	15	10	11	10	9	7
		3	63	9	11		10				11			9		
		4	84	12	21	19		17		16	19	14	17		16	
	5	5	105	15	36	32	31	26		27	35	29	28	23	26	23
		6	126	18	51	43	44	39	40	38	51	41	40	39	35	38
		7	147	21	67	55	56	50	51	45	64	53		51	48	50
		3	72	3				3						3		
		4	96	4	5			4						4		
	5	5	120	5	9	6			5		7			5		
		6	144	6	14		9	8		7	10	7			6	
5		7	168	7	24	17		11		10		11	8	9	10	7
		3	72	9	12		10		9		12		10		9	
		4	96	12	24	18	19	17	16	18	21	18		17		15
	6	5	120	15	39	31	29		25	30	38	31	28	27	26	21
		6	144	18	53	50	46	47	42	39	53	41	44	42		
		7	168	21	78	65	62	59	60	58	72	61	57	56	51	54

Table 4: Using the CM to estimate the size of a d-min-cut of a SW(mult,D) graph for large topologies.

equal to 3, independently of the processors concurrently run (i.e., 1, 64, 128, 256, 512, and 1024), or the number of trails assigned to each processor (10^6 or 5.10^6). The green colored cells correspond to cells in which the estimates are equal to the exact values of $s_{min}(SW(mult,D))$.

As it can be noted, the estimates were more accurate whenever the graph has small $s_{min}(SW(mult,D))$ value (i.e., *d*-connectivity). For example the exact *d*-connectivity was correctly estimated (i.e., $s_{min}(SW(mult,D))=mult$) when $d=D$, and less accurate whenever $d>D$ (i.e., $s_{min}(SW(mult,D))=3.mult$). Increasing the number of trials per processor by a factor of 5 improved the accuracy of the estimates by a constant value for the tests conducted on the same number of processors. For example for $D=4$, $mult=5$, and $d=5$, the estimated size of the *d*-min-cut went from 27 to 23 (on 1024 processors) with exact *d*-connectivity of 15. The results show that the accuracy of the PCM using parallel processing is more significant when trying to calculate the *d*-connectivity of a family of graphs with low -edge - connectivity.

Among several family of graphs with low *d*-connectivity are for example the *k*-dimensional Hypercubes H_k, having a maximum degree *k*. Other examples are regular graphs (i.e., all the vertices have the same degree) of fixed degree *k*; examples of the latter are cubic-graphs, in which all the vertices are of degree 3 (e.g., Dodecahedron Petersen [16], etc.), or circulant graphs [17] with a fixed number of *k* jumps (these graphs have vertices of degree 2.*k*).

Conclusion and Future Work

In this work, MC Construction methods were combined with parallel processing computing to give estimates on the number of *d*-path-sets with specific edge cardinality, and, as a by-product, estimates for the Diameter Constrained Reliability measure of a network. Tests conducted on different topologies yielded accurate estimates especially when calculating the number of *d*-path-sets composed of *j* edges, and *j* approaches $m=|E|$. This fact suggested a further analysis for estimating the size of *d*-min-cuts of graphs are required. Tests for estimating *d*-min-cuts showed that the accuracy of the methodology was more significant on low connectivity graphs.

Acknowledgments

This research was supported by the City University of New York TRADB-42-426 Research Award.

References

1. Petingi L, Rodriguez J (2001) Reliability of networks with delay constraints. Congressus Numerantium 152: 117-123.
2. Petingi L (2013) Diameter-related properties of graphs and applications to network reliability theory. WSEAS Transactions on Math, 12: 884-894.
3. Petingi L, Rodriguez J (2000) Bounds on the maximum number of edge-disjoint steiner trees of a graph. Congressus Numerantium, 145: 43-52.
4. Deering S, Hinden R (1998) Rfc 2460, internet protocol, version 6 (ipv6) specification.
5. Ball MO (1986) Computational complexity of network reliability analysis: An overview. IEEE Trans Reliab 35: 230-239.
6. Elperin T, Gertsbakh I, Lomonosov M (1991) Estimation of network reliability using graph evolution model. IEEE Trans Reliab 40: 572-581.
7. Getsbakh IB (2000) Reliability Theory With Applications to Preventive Maintenance. Springer.
8. Gertsbakh IB, Shpungin Y (2009) Models of Network Reliability: Analysis, Combinatorics and Monte Carlo. CRC Press.
9. Gertsbakh IB, Shpungin Y (2011) Network Reliability and Resilience. Springer Briefs in Electrical and Computer Engineering, Springer.
10. Colbourn CJ (1987) The Combinatorics of Network Reliability. Oxford University Press.
11. Gertsbakh B, Petingi L, Wojcik S (2017) Construction/destruction monte carlo to estimate the number of path-sets for the reliability measure of a network with diameter constraints. Int'l Conf Scientific Computing (CSC'17) 57-61.
12. Floyd RW (1962) Algorithm 97: Shortest path Comm ACM 5: 345.
13. Quinn M J (2003) Parallel Programming in C with MPI and OpenMP. McGraw-Hill.
14. Ford LR, Fukerson DR (1962) Flows in Networks. Princeton University Press.
15. Mahjoub AR, McCornick ST (2010) Max flow and min cut with bounded-length paths: Complexity, algorithms, and approximations. Math Program Ser B 124: 271-284.
16. Harary F (1969) Graph Theory. Addison-Wesley series in mathematics. Addison-Wesley Publishing Company, Boston, 1969.
17. Evodkimov SA, Ponomarenko IN (2004) Circulant graphs: Recognizing and isomorphism testing in polynomial time. St Petersburg Math J 15: 813-835.