

Minimizing Transmission Time under Required System Capacity Constraint in Stochastic-Flow Networks

Moatamad Refaat Hassan

Computer Science Branch, Department of Mathematics,
Faculty of Science, Aswan University, Aswan, Egypt

Article history

Received: 06-12-2017

Revised: 02-04-2018

Accepted: 04-05-2018

Email: m_r_hassan73@yahoo.com

Abstract: In the case of an existing limited capacity of the system called required system capacity, in this study a new method is presented to determine the disjoint paths to send data from the source to the sink. System reliability is evaluated as the probability that a specified amount of data can be transmitted from a source to a sink through two disjoint paths, subject to a time threshold and a required capacity. In addition, we study the effect of using the proposed method on the reliability evaluation in comparison with other previous methods.

Keywords: Quickest Path Problem, Disjoint Paths, Stochastic-Flow Network (SFN)

Introduction

The QPP is used to find a path that has the minimum transmission time to send a specified amount of data from the source to the sink node (Chin and Chen, 1990; Chen and Hung, 1993). All pairs quickest problem has been solved in (Lee and Papadopoulou, 1993). If the quickest paths are required to go through a specified path, then the restricted problem is called the constrained QPP (Chen and Hung, 1994). Taking into account the lead-time and the capacity of each arc (Martins and Santos, 1997), the problem is solved as a bicriteria path problem, (Sedeno-Noda and Gonzalez-Barrera, 2014). In addition, Lin (2009a) proposed a method based on Monte-Carlo simulation to solve the quickest path flow network reliability problem.

The QPP has been extended to the stochastic-flow network by (Lin, 2003), an algorithm proposed to evaluate the system reliability of a SFN with time consideration. While in (Lin, 2009a), the system reliability has been evaluated under time constraint. In this case, the data are transmitted through two disjoint paths. Lin (2009b) presented an algorithm to determine the optimal routing policy with the highest system reliability. Lin (2010) proposed an algorithm to generate all minimal system states fulfilling the demand, time and budget constraints; then the system reliability is evaluated in terms of such system states. The network reliability has been evaluated in the case of sending units of data through a number of MPs simultaneously under both time and budget constraints (Lin, 2011a). Moreover, they evaluated network reliability according

to the spare routing. In order to reduce transmission time, in (Lin, 2011b), a problem of transmitting data through multiple disjoint minimal paths simultaneously was presented. In (Yeh, 2015), an algorithm based on depth-first-search to solve the quickest path reliability problem was proposed. In addition, in (El Khadiri and Yeh, 2016), a method based on Monte-Carlo simulation to solve the quickest path flow network reliability problem was proposed. The required system capacity (Hassan, 2012) has been extended in (Aggarwal *et al.*, 1982) to solve the QPP in a stochastic flow network. Furthermore, the system reliability has been evaluated under time threshold and system capacity constraint.

Previous studies have evaluated the system reliability of an SFN network according to a given demand under the time constraints (Sedeno-Noda and Gonzalez-Barrera, 2014; Lin, 2003; 2009a; 2009b; 2010; 2011a; 2011b; Yeh, 2015; El Khadiri and Yeh, 2016; Yeh *et al.*, 2013; Jane and Laih, 2017). However, these studies have not considered the case of the required system capacity or the system with limited capacity. Therefore, the main purpose of this paper is to study the QPP in an SFN in the case of sending data simultaneously through two DMPs under required system capacity.

The required system capacity has been used to determine the available disjoint paths to send a specified amount of data from the source node to the sink node. Hereby, the set of solutions can be obtained and the system reliability can then be computed in terms of these solutions. The algorithm presented by Lin (2003) can be modified to exploit the idea of using the required system capacity constraint to

determine the available disjoint paths. The same technique can be used to obtain the set of solutions.

The paper is organized as follows. In section II, the formulation of the problem is presented. System reliability evaluation is given in section III. In section IV, the proposed algorithm to generate all solutions, called Algorithm I is presented. In section V, illustrative examples are presented to show the efficiency of the proposed algorithm. The computational time of the proposed algorithm is presented in section VI. An important Lemma that can be used to speed up Algorithm I is given in section VII. In section VIII, both the discussion and comparison are presented. In section IX, we provide the conclusion.

Problem formulation

The transmission time TP_j of a path P_j is given by:

$$TP_j = \sum_{k=1}^n \{l_k | a_k \in P_j\} + \lceil d/v \rceil \quad (1)$$

where, v is the smallest integer such that $TP_j \leq T$, T is the time threshold.

The maximum capacity CP_j of a path P_j is given by:

$$CP_j = \text{Min}\{M^k | a_k \in P_j\} \quad (2)$$

The path P_j is said to be the available path under the required system capacity if its maximum capacity is greater than or equal to the required system capacity, i.e., when the condition $CP_j \geq Cs$ is satisfied. Then the set of available paths ρ is given by:

$$\rho = \{P_j | CP_j \geq Cs, j = 1, 2, \dots, np\} \quad (3)$$

If no path satisfies the condition $CP_j \geq Cs$, then $\rho = \emptyset$.

From the set ρ , we can construct the set of disjoint paths ϕ , where:

$$\phi = \{(P_i, P_j) \in \rho | i \neq j \text{ and } P_i \cap P_j = \emptyset\} \quad (4)$$

The transmission time of sending d units of data simultaneously through two disjoint paths P_i and P_j under capacity vector X is $\psi(d_i, X, P_i)$ and $\psi(d_j, X, P_j)$ respectively, where $d_i + d_j = d$ such that $CP_i \geq Cs$ and $CP_j \geq Cs$. Then, $\psi(d_i, X, P_i) = \sum_{k=1}^n l_k | a_k \in P_i + \lceil \frac{d_i}{Cs} \rceil$ and $\psi(d_j, X, P_j) = \sum_{k=1}^n l_k | a_k \in P_j + \lceil \frac{d_j}{Cs} \rceil$. Let Γ be the set of candidate solutions X under the above conditions and let $\Gamma_{\min} = \{X | X \text{ is a minimal line in } \Gamma\}$. Then, $X \in \Gamma_{\min}$ is called a (d, T, Cs) -DMPs throughout this paper. For simplicity

purposes, let T_i and T_j be denoted to $\psi(d_i, X, P_i)$ and $\psi(d_j, X, P_j)$ respectively.

Reliability Evaluation

If X^1, X^2, \dots, X^b are all (d, T, Cs) -DMPs obtained, then the system reliability $R_{d,T,Cs}$ is given by:

$$R_{d,T,Cs} = \Pr\left\{\bigcup_{i=1}^b \{Y | Y \geq X^i\}\right\} \quad (5)$$

The above formula can be calculated by the inclusion-exclusion rule (Janan, 1985; Locks, 1980) state space decomposition (Alexopoulos, 1995) and Recursive Sum of Disjoint Products (RSDP) (Zuo *et al.*, 2007), used in this study.

Proposed Algorithm to Generate all (d, T, Cs) -DMPs

The following algorithm uses required value of the Cs to determine the available MPs and then generate all (d, T, Cs) -DMPs.

Input: A stochastic-flow network and minimal paths, demand level, time limitation and system capacity constraint.

Output: All the (d, T, Cs) -DMPs.

Algorithm I: Generate all (d, T, Cs) -DMPs.

1. Input the paths $P_i, i = 1, 2, 3, \dots, \omega$, demand value and the Cs value.
2. Calculate the minimum capacity for each path $mcp_i, CP_i = \text{Min}\{M_k | a_k \in P_i\}$.
3. Determine all paths that satisfy the condition $CP_i \geq Cs$.
4. Construct the set of available disjoint paths $\phi = \{(P_i, P_j) | i \neq j \text{ and } P_i \cap P_j = \emptyset\}$ from the paths generated in step 3.
5. For each disjoint pair P_i and P_j in ϕ do the following:
 - 5.1. Find the largest demands \bar{d}_i and \bar{d}_j that can be assigned to P_i and P_j respectively. Such that:

$$\sum_{k=1}^n \{l_k | a_k \in P_i\} + \lceil \bar{d}_i / Cs \rceil \leq T$$

and:

$$\sum_{k=1}^n \{l_k | a_k \in P_j\} + \lceil \bar{d}_j / Cs \rceil \leq T$$

- 5.2. Generate the set of solutions:

$$\beta = \{(d_i, d_j) | d_i \leq \bar{d}_i, d_j \leq \bar{d}_j \text{ and } d_i + d_j = d\}$$

5.3. For all solutions in β calculate the transmission time for each path:

$$T_i = \sum_{k=1}^n \{l_k \mid a_k \in P_i\} + \lceil d_i / C_s \rceil$$

and:

$$T_j = \sum_{k=1}^n \{l_k \mid a_k \in P_j\} + \lceil d_j / C_s \rceil$$

5.4. If $T_i \leq T$ and $T_j \leq T$, then generate the capacity vector $X^q = (x_1, x_2, \dots, x_k, \dots, x_n)$ as follows:

$$x_k = \begin{cases} C_s & \text{if } a_k \in P_i \cup P_j \\ 0 & \text{Otherwise} \end{cases}$$

5.5. Repeat steps 5.3 and 5.4 to generate all solutions.

5.6. Remove the solutions that are not lower.

6. Evaluate the system reliability for the generated lower capacity vectors.

7. End

Illustrative Examples

A. Five Nodes Network Example

The network of Fig. 1 has five nodes and eight arcs, (Lin, 2009a). The information of each arc is given in Table 1. The Minimal paths and their maximum capacity are given in Table 2. The available disjoint minimal paths corresponding to each value for C_s , the available values for (d_1, d_2) , the corresponding (d, T, C_s) -DMP solutions and R_{d, T, C_s} are given in Table 3. The best solutions found are given in Table 4. The demand d equals to 8 and the time limit is 9 for this example.

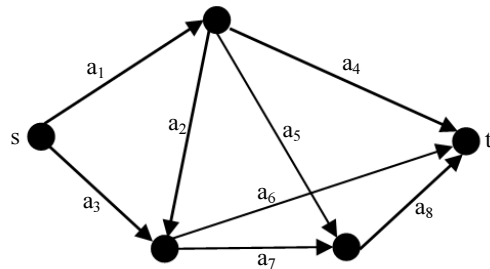


Fig. 1: Computer network

Table 1: Arcs information

Arc	Capacity	Probability	Lead time	Arc	Capacity	Probability	Lead time
a ₁	0	0.05	2	a ₅	0	0.10	1
	1	0.05			1	0.10	
	2	0.10			2	0.80	
	3	0.80					
a ₂	0	0.05	1	a ₆	0	0.05	2
	1	0.05			1	0.05	
	2	0.10			2	0.10	
	3	0.80			3	0.20	
a ₃	0	0.05	3	a ₇	4	0.60	2
	1	0.05			0	0.05	
	2	0.10			1	0.10	
	3	0.80			2	0.10	
a ₄	0	0.10	3	a ₈	3	0.10	1
	1	0.90			4	0.10	
					5	0.55	
					0	0.05	
					1	0.05	
					2	0.10	
					3	0.10	
					4	0.70	

Table 2: The paths and their corresponding maximum capacity

No.	P_i	CP_i
1	{a ₁ , a ₄ }	1
2	{a ₁ , a ₅ , a ₈ }	2
3	{a ₁ , a ₂ , a ₆ }	3
4	{a ₁ , a ₂ , a ₇ , a ₈ }	3
5	{a ₃ , a ₆ }	3
6	{a ₃ , a ₇ , a ₈ }	3

Table 3: The results for different values of Cs

Cs	Disjoint pair	(d ₁ ,d ₂)	The capacity vector X	R _{d,T,Cs}
1	P ₁ = {a ₁ , a ₄ } and P ₅ = {a ₃ , a ₆ }	(4,4)	1 0 1 1 0 1 0 0	0.771638
	P ₁ = {a ₁ , a ₄ } and P ₆ = {a ₃ , a ₇ , a ₈ }	*	-	-
2	P ₂ = {a ₁ , a ₅ , a ₈ } and P ₅ = {a ₃ , a ₆ }	(4,4)	1 0 1 0 1 1 0 1	0.733056
		(8,0)	2 0 0 0 2 0 0 2	0.93312
		(7,1)	2 0 2 0 2 2 0 2	
		(6,2)	2 0 2 0 2 2 0 2	
		(5,3)	2 0 2 0 2 2 0 2	
		(4,4)	2 0 2 0 2 2 0 2	
		(3,5)	2 0 2 0 2 2 0 2	
		(2,6)	2 0 2 0 2 2 0 2	
		(1,7)	2 0 2 0 2 2 0 2	
	P ₃ = {a ₁ , a ₂ , a ₆ } and P ₆ = {a ₃ , a ₇ , a ₈ }	(0,8)	0 0 2 0 0 2 0 0	
		(8,0)	2 2 0 0 0 2 0 0	
		(7,1)	2 2 2 0 0 2 2 2	
		(6,2)	2 2 2 0 0 2 2 2	
		(5,3)	2 2 2 0 0 2 2 2	
		(4,4)	2 2 2 0 0 2 2 2	
		(3,5)	2 2 2 0 0 2 2 2	
		(2,6)	2 2 2 0 0 2 2 2	
		(1,7)	2 2 2 0 0 2 2 2	
3	P ₄ = {a ₁ , a ₂ , a ₇ , a ₈ } and P ₅ = {a ₃ , a ₆ }	(6,2)	2 2 2 0 0 2 2 2	0.81
		(5,3)	2 2 2 0 0 2 2 2	
		(4,4)	2 2 2 0 0 2 2 2	
		(3,5)	2 2 2 0 0 2 2 2	
		(2,6)	2 2 2 0 0 2 2 2	
		(1,7)	2 2 0 0 2 2 2	
	P ₃ = {a ₁ , a ₂ , a ₆ } and P ₆ = {a ₃ , a ₇ , a ₈ }	(0,8)	0 0 2 0 0 2 0 0	0.74624
		(8,0)	3 3 0 0 0 3 0 0	
		(7,1)	3 3 3 0 0 3 3 3	
		(6,2)	3 3 3 0 0 3 3 3	
		(5,3)	3 3 3 0 0 3 3 3	
		(4,4)	3 3 3 0 0 3 3 3	
3	P ₄ = {a ₁ , a ₂ , a ₇ , a ₈ } and P ₅ = {a ₃ , a ₆ }	(3,5)	3 3 3 0 0 3 3 3	0.77824
		(2,6)	3 3 3 0 0 3 3 3	
		(1,7)	3 3 3 0 0 3 3 3	
		(0,8)	0 0 3 0 0 0 3 3	
		(8,0)	3 3 0 0 0 0 3 3	
		(7,1)	3 3 3 0 0 3 3 3	
	P ₃ = {a ₁ , a ₂ , a ₆ } and P ₆ = {a ₃ , a ₇ , a ₈ }	(6,2)	3 3 3 0 0 3 3 3	0.77824
		(5,3)	3 3 3 0 0 3 3 3	
		(4,4)	3 3 3 0 0 3 3 3	
		(3,5)	3 3 3 0 0 3 3 3	
		(2,6)	3 3 3 0 0 3 3 3	
		(1,7)	3 3 3 0 0 3 3 3	
	(0,8)	0 0 3 0 0 3 0 0		

*Means that (d₁,d₂) does not satisfy one or more conditions mentioned in the algorithm. Also for the disjoint pairs {P₃,P₆} and {P₄, P₅}

Table 4: The best disjoint paths for each Cs value

Cs	The best disjoint pair	R _{d,T,Cs}
1	P ₁ = {a ₁ , a ₄ } and P ₅ = {a ₃ , a ₆ }	0.771638
2	P ₂ = {a ₁ , a ₅ , a ₈ } and P ₅ = {a ₃ , a ₆ }	0.933120
3	P ₄ = {a ₁ , a ₂ , a ₇ , a ₈ } and P ₅ = {a ₃ , a ₆ }	0.778240

B. Fourteen Nodes Network Example

The network given in Fig. 2 is taken from (Lin, 2011a). The arcs' information is given in Table 5. The network

contains eleven paths shown in Table 6. Also, the maximal capacity for each path has been determined. The available paths for different values of Cs are shown in Table 7. The best disjoint pair of paths for each Cs shown in Table 8.

Table 5: Arcs information

Arc	Capacity	Probability	Lead time	Arc	Capacity	Probability	Lead time
a ₁	0	0.06	2	a ₁₂	0	0.04	2
	10	0.03			10	0.05	
	30	0.05			20	0.04	
	50	0.86			40	0.04	
a ₂	0	0.05	2	a ₁₃	60	0.83	1
	10	0.02			0	0.04	
	30	0.03			10	0.02	
	50	0.9			20	0.02	
a ₃	0	0.04	3	a ₁₄	40	0.01	2
	10	0.04			60	0.82	
	20	0.04			0	0.05	
	40	0.88			20	0.95	
a ₄	0	0.05	3	a ₁₅	0	0.05	3
	10	0.05			10	0.05	
	30	0.05			30	0.05	
	50	0.85			50	0.05	
a ₅	0	0.05	4	a ₁₆	70	0.8	2
	10	0.05			0	0.04	
	30	0.05			10	0.04	
	50	0.85			30	0.04	
a ₆	0	0.04	3	a ₁₇	40	0.06	3
	10	0.04			60	0.82	
	20	0.04			0	0.05	
	40	0.88			10	0.04	
a ₇	0	0.04	2	a ₁₈	30	0.03	3
	20	0.96			50	0.88	
					0	0.05	
					10	0.05	
a ₈	0	0.05	3	a ₁₉	30	0.05	3
	10	0.05			40	0.88	
	30	0.05			0	0.05	
	50	0.85			10	0.05	
a ₉	0	0.05	4	a ₂₀	30	0.03	3
	10	0.05			50	0.9	
	20	0.07			0	0.05	
	40	0.83			10	0.05	
a ₁₀	0	0.05	2	a ₂₁	20	0.1	2
	10	0.05			40	0.8	
	20	0.05			0	0.07	
	40	0.85			20	0.93	
a ₁₁	0	0.05	3	a ₂₂	0	0.04	4
	10	0.05			10	0.96	
	30	0.02					
	50	0.88					

Table 6: The paths and their maximum capacity

No.	P _i	CP _i
1	{a ₁ , a ₂ , a ₃ }	40
2	{a ₁ , a ₆ , a ₇ }	20
3	{a ₄ , a ₅ , a ₆ }	40
4	{a ₈ , a ₉ , a ₁₀ }	40
5	{a ₁₀ , a ₁₁ , a ₁₄ }	20
6	{a ₁₁ , a ₁₂ , a ₁₃ }	50
7	{a ₁₅ , a ₂₂ }	10
8	{a ₁₅ , a ₁₆ , a ₁₇ }	50
9	{a ₁₈ , a ₂₁ , a ₂₂ }	10
10	{a ₁₆ , a ₁₇ , a ₁₈ , a ₂₁ }	20
11	{a ₁₈ , a ₁₉ , a ₂₀ }	40

Table 7: The available paths for different values of Cs

Cs	P _i	Cs	P _i	Cs	P _i
50	{a ₁₁ , a ₁₂ , a ₁₃ }	20	{a ₁₁ , a ₁₂ , a ₁₃ }	10	{a ₁ , a ₂ , a ₃ }
	{a ₁₅ , a ₁₆ , a ₁₇ }		{a ₁₅ , a ₁₆ , a ₁₇ }		{a ₁ , a ₆ , a ₇ }
40	{a ₁ , a ₂ , a ₃ }		{a ₄ , a ₅ , a ₆ }		{a ₄ , a ₅ , a ₆ }
	{a ₈ , a ₉ , a ₁₀ }		{a ₈ , a ₉ , a ₁₀ }		
	{a ₁₈ , a ₁₉ , a ₂₀ }		{a ₁₀ , a ₁₁ , a ₁₄ }		
			{a ₁₁ , a ₁₂ , a ₁₃ }		
			{a ₁₅ , a ₂₂ }		
			{a ₁₅ , a ₁₆ , a ₁₇ }		
			{a ₁₈ , a ₂₁ , a ₂₂ }		
			{a ₁₆ , a ₁₇ , a ₁₈ , a ₂₁ }		
			{a ₁₈ , a ₁₉ , a ₂₀ }		

Table 8: The best disjoint paths for each Cs

Cs	The best disjoint pair	R _{d,T,Cs}
10	P ₁ = {a ₁ , a ₂ , a ₃ } and P ₆ = {a ₁₁ , a ₁₂ , a ₁₃ }	0.91
20	P ₁ = {a ₁ , a ₂ , a ₃ } and P ₆ = {a ₁₁ , a ₁₂ , a ₁₃ }	0.91
40	P ₁ = {a ₁ , a ₂ , a ₃ } and P ₆ = {a ₁₁ , a ₁₂ , a ₁₃ }	0.822451
50	P ₆ = {a ₁₁ , a ₁₂ , a ₁₃ } and P ₈ = {a ₁₅ , a ₁₆ , a ₁₇ }	0.789727

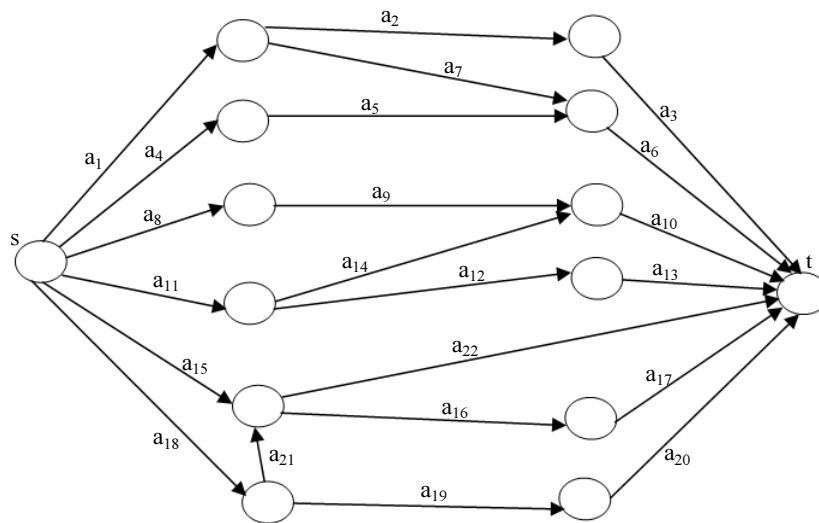


Fig. 2: Example of a Network of fourteen nodes

Time Analysis

The algorithm Computes the maximum capacity of each path in $O(\omega n)$ time and $O(\omega)$ to determine the set of disjoint paths. The time needed to determine \bar{d}_1 and \bar{d}_2 is $O(\alpha n)$, where α represents the number of disjoint paths. To test transmission time, it takes $O(\alpha n)$. To generate all (d, T, Cs) -DMPs, it takes $O(\alpha dn)$ time. Then, the total time is $O(\delta n)$ taken by the proposed algorithm to find Γ_{\min} , where $\delta = \omega + (2 + d)\alpha$.

An Important Lemma

The time needed to compare each candidate solution X with all other solutions to construct Γ_{\min} is $O(dn)$, in the worst case scenario. This time can be decreased by using the following Lemma.

Lemma 1

If X is (d, T) -DP candidate under Cs and X corresponds to the solution $(d_1, 0)$ or $(0, d_2)$, then X is a lower vector.

Table 9: The lower vectors in example A according to Lemma 1

Cs	The disjoint pair	(d ₁ ,d ₂)	The lower vectors	R _{d,T,Cs}
2	P ₂ = {a ₁ , a ₅ , a ₈ } and P ₅ = {a ₃ , a ₆ }	(8,0)	2 0 0 0 2 0 0 2	0.93312
		(0,8)	0 0 2 0 0 2 0 0	
	P ₃ = {a ₁ , a ₂ , a ₆ } and P ₆ = {a ₃ , a ₇ , a ₈ }	(8,0)	2 2 0 0 0 2 0 0	0.729
		(8,0)	0 0 2 0 0 2 0 0	0.81
3	P ₃ = {a ₁ , a ₂ , a ₆ } and P ₆ = {a ₃ , a ₇ , a ₈ }	(8,0)	3 3 0 0 0 3 0 0	0.74624
		(0,8)	0 0 3 0 0 0 3 3	
	P ₄ = {a ₁ , a ₂ , a ₇ , a ₈ } and P ₅ = {a ₃ , a ₆ }	(8,0)	3 3 0 0 0 0 3 3	0.77824
		(8,0)	0 0 3 0 0 0 3 0	
		(0,8)	0 0 3 0 0 0 3 0	

Proof

Let P₁ = {a₁, a₄} and P₂ = {a₃, a₆} and Cs = V. Let X¹ = (V 0 0 V 0 0) and X² = (0 0 V 0 0 V) are (d, T)-DMP candidate to (d1,0) and (0, d2) respectively. Then all other (d, T)-DMP candidates will be in the form (V 0 VV 0 V). It is easy to verify that X¹ and X² are lower. Table 9 shows lower vectors based on using this Lemma in Example A.

Discussion and Comparison

In this section, we investigate how to use the system capacity constraint to determine the candidate disjoint paths and then use Lin's Algorithm presented in (Lin, 2009a) to obtain the solutions, given by Algorithm II. In addition, the results of applying Algorithm II on both Example A and Example Bare shown in Table 10 to 12. While Table 13 gives the comparison between the results obtained by the Algorithm I and that obtained by Algorithm II to show the difference to the reliability without using Cs value to deduce the solutions.

Algorithm II: Using Cs to the available MPs and using Lin's Algorithm to generate all (d, T)-DMPs.

1. Input the paths P_i, i = 1,2,3,..., ω and the Cs value.
2. Calculate the maximum capacity for each path CP_i, CP_j = Min{M_k | a_k ∈ P_j}.
3. Determine all paths that satisfy the condition CP_i ≥ Cs.
4. Construct the set of available disjoint paths φ = {(P_i, P_j) | i ≠ j and P_i ∩ P_j = ∅} from the paths generated in step 3.
5. For each disjoint pair P_i and P_j in φ, do the following (Lin, 2009a):
 - 5.1. Find the largest demands \bar{d}_i and \bar{d}_j that can be assigned to P_i and P_j respectively, such that:

$$\sum_{k=1}^n \{l_k | a_k \in P_i\} + \lceil \bar{d}_i / CP_i \rceil \leq T$$

and:

$$\sum_{k=1}^n \{l_k | a_k \in P_j\} + \lceil \bar{d}_j / CP_j \rceil \leq T .$$

- 5.2. Generate the set of solutions $\beta = \{(d_i, d_j) | d_i \leq \bar{d}_i, d_j \leq \bar{d}_j \text{ and } d_i + d_j = d\}$
- 5.3. For all solutions in β, find the smallest integers v_i and v_j, such that the transmission time for each path is calculated as:

$$\sum_{k=1}^n \{l_k | a_k \in P_i\} + \lceil d_i / v_i \rceil \leq T$$

and:

$$\sum_{k=1}^n \{l_k | a_k \in P_j\} + \lceil d_j / v_j \rceil \leq T .$$

- 5.4. Generate the capacity vector X^q = (x₁, x₂, ..., x_k, ..., x_n) as follows:

$$x_k = \begin{cases} u_k \geq v_i & \text{if } a_k \in P_i \\ u_k \geq v_j & \text{if } a_k \in P_j \\ 0 & \text{Otherwise} \end{cases}$$
- 5.5. Repeat steps 5.3 and 5.4 to generate all solutions.
- 5.6. Remove the solutions that are not lower.
6. Evaluate the system reliability for the generated lower capacity vectors.
7. End

The results in Table 13 show that the system reliability values are affected by using Cs value to generate solutions. The results obtained by using both Lin's Algorithm and Algorithm II are identical. However, Algorithm II can be used to determine the available disjoint paths and then generate the solutions.

Table 10: The results of applying algorithm II to example A

Cs	The disjoint pair	$R_{d,T,Cs}$
1	$P_1 = \{a_1, a_4\}$ and $P_5 = \{a_3, a_6\}$	0.889088
	$P_1 = \{a_1, a_4\}$ and $P_6 = \{a_3, a_7, a_8\}$	0.658268
	$P_2 = \{a_1, a_5, a_8\}$ and $P_5 = \{a_3, a_6\}$	0.948313
	$P_3 = \{a_1, a_2, a_6\}$ and $P_6 = \{a_3, a_7, a_8\}$	0.885846
2	$P_4 = \{a_1, a_2, a_7, a_8\}$ and $P_5 = \{a_3, a_6\}$	0.904758
	$P_2 = \{a_1, a_5, a_8\}$ and $P_5 = \{a_3, a_6\}$	0.948313
	$P_3 = \{a_1, a_2, a_6\}$ and $P_6 = \{a_3, a_7, a_8\}$	0.885846
	$P_4 = \{a_1, a_2, a_7, a_8\}$ and $P_5 = \{a_3, a_6\}$	0.904758
3	$P_3 = \{a_1, a_2, a_6\}$ and $P_6 = \{a_3, a_7, a_8\}$	0.885846
	$P_4 = \{a_1, a_2, a_7, a_8\}$ and $P_5 = \{a_3, a_6\}$	0.904758

Table 11: The best disjoint paths for each Cs value using algorithm II To Example A

Cs	The best disjoint pair	$R_{d,T,Cs}$
1	$P_2 = \{a_1, a_5, a_8\}$ and $P_5 = \{a_3, a_6\}$	0.948313
2	$P_2 = \{a_1, a_5, a_8\}$ and $P_5 = \{a_3, a_6\}$	0.948313
3	$P_4 = \{a_1, a_2, a_7, a_8\}$ and $P_5 = \{a_3, a_6\}$	0.904758

Table 12: The best disjoint paths for each Cs value using algorithm II to example B.

Cs	The best disjoint pair	$R_{d,T,Cs}$
10	$P_1 = \{a_1, a_2, a_3\}$ and $P_6 = \{a_{11}, a_{12}, a_{13}\}$	0.841395
20	$P_1 = \{a_1, a_2, a_3\}$ and $P_6 = \{a_{11}, a_{12}, a_{13}\}$	0.841395
40	$P_1 = \{a_1, a_2, a_3\}$ and $P_6 = \{a_{11}, a_{12}, a_{13}\}$	0.841395
50	$P_6 = \{a_{11}, a_{12}, a_{13}\}$ and $P_8 = \{a_{15}, a_{16}, a_{17}\}$	0.834777

Table 13: Comparison results

Example A			Example B		
Cs	Reliability using Algorithm I	Reliability using Algorithm II	Cs	Reliability using Algorithm I	Reliability using Algorithm II
1	0.771638	0.841395	10	0.91	0.841395
2	0.93312	0.841395	20	0.91	0.841395
3	0.77824	0.841395	40	0.822451	0.841395
			50	0.789727	0.834777

Conclusion

Taking into account the required system capacity (C_s) and the transmission time (T), the paper presented an algorithm to determine the set of disjoint paths that would simultaneously carry the data and then, generate all (d, T, C_s) -DMPs to calculate the system reliability $R_{(d,T,C_s)}$. Moreover, in this study, the benefits of using the system capacity to determine the available paths that can be used to send data have been investigated. In addition, the computational complexity taken by the algorithm to generate all solutions is $O(\delta n)$.

Acknowledgment

We would like to thank all the parties involved in this research work.

Ethics

The author confirm that they have thoroughly seen the content of the paper and do not find any conflict of interest and ethical issues.

References

- Aggarwal, K.K., Y.C. Chopra and J.S. Bajwa, 1982. Capacity consideration in reliability analysis of communication system. IEEE Trans. Reliability, 31: 177-181. DOI: 10.1109/TR.1982.5221290
- Alexopoulos, C., 1995. A Note on state-space decomposition methods for analyzing stochastic flow networks. IEEE Trans. Reliability, 44: 354-357. DOI: 10.1109/24.387394
- Chen, G.H. and Y.C. Hung, 1993. On the quickest path problem. Inform. Process. Lett., 46: 125-128. DOI: 10.1016/0020-0190(93)90057-G

- Chen, G.H. and Y.C. Hung, 1994. Algorithms for the constrained quickest path problem and the enumeration of quickest paths. *Comput. Operat. Res.*, 21: 113-118.
 DOI: 10.1016/0305-0548(94)90045-0
- Chin, Y.H. and Y.L. Chen, 1990. The quickest path problem. *Comput. Operat. Res.*, 17: 153-161.
 DOI: 10.1016/0305-0548(90)90039-A
- El Khadiri, M. and W.C. Yeh, 2016. An efficient alternative to the exact evaluation of the quickest path flow network reliability problem. *Comput. Operat. Res.*, 76: 22-32.
 DOI: 10.1016/j.cor.2016.06.010
- Hassan, M.R., 2012. Reliability evaluation of stochastic-flow network under quickest path and system capacity constraints. *Int. J. Comput. Netw.*, 4: 98-103.
- Janan, X., 1985. On multistate system analysis. *IEEE Trans. Reliability*, r-34: 329-337.
- Jane, C.C. and Y.W. Lai, 2017. Distribution and reliability evaluation of max-flow in dynamic multi-state flow networks. *Eur. J. Operat. Res.*, 259: 1045-1053. DOI: 10.1016/j.ejor.2016.12.030
- Lee, D.T. and E. Papadopoulou, 1993. The all-pairs quickest path problem. *Inform. Process. Lett.*, 45: 261-367. DOI: 10.1016/0020-0190(93)90214-T
- Lin, Y.K., 2003. Extend the quickest path problem to the system reliability evaluation for a stochastic-flow network. *Comput. Operat. Res.*, 30: 567-575.
 DOI: 10.1016/S0305-0548(02)00025-4
- Lin, Y.K., 2009a. Time version of the shortest path problem in a stochastic-flow network. *J. Comput. Applied Math.*, 228: 150-157.
 DOI: 10.1016/j.cam.2008.09.003
- Lin, Y.K., 2009b. Routing policy of stochastic-flow networks under time threshold and budget constraint. *Expert Syst. Applic.*, 36: 6076-6081.
 DOI: 10.1016/j.eswa.2008.07.080
- Lin, Y.K., 2010. Reliability of k separate minimal paths under both time and budget constraints. *IEEE Trans. Reliability*, 59: 183-190.
 DOI: 10.1109/TR.2009.2035794
- Lin, Y.K., 2011a. Network reliability of a time-based multistate network under spare routing with ρ minimal paths. *IEEE Trans. Reliability*, 60: 61-69.
 DOI: 10.1109/TR.2010.2103594
- Lin, Y.K., 2011b. Stochastic flow networks via multiple paths under time threshold and budget constraint. *Comput. Math. Applic.*, 62: 2629-2638.
 DOI: 10.1016/j.camwa.2011.08.002
- Locks, M.O., 1980. Recursive disjoint products, inclusion-exclusion and min-cut approximations. *IEEE Trans. Reliability*, R-29: 425-436.
 DOI: 10.1109/TR.1980.5220885
- Martins, E.D.Q.V. and J.L.E.D. Santos, 1997. An algorithm for the quickest path problem. *Operat. Res. Lett.*, 20: 195-198.
 DOI: 10.1016/S0167-6377(97)00008-4
- Sedeno-Noda, A. and J.D. Gonzalez-Barrera, 2014. Fast and fine quickest path algorithm. *Eur. J. Operat. Res.*, 238: 596-606.
 DOI: 10.1016/j.ejor.2014.04.028
- Yeh, W.C., 2015. A fast algorithm for quickest path reliability evaluations in multi-state flow networks. *IEEE Trans. Reliability*, 64: 1175-1184.
 DOI: 10.1109/TR.2015.2452580
- Yeh, W.C., L.E. Lin, Y.C. Chou and Y.C. Chen, 2013. Optimal routing for multi-commodity in multistate flow network with time constraints. *Qual. Technol. Quant. Manage.*, 10: 161-177.
 DOI: 10.1080/16843703.2013.11673314
- Zuo, M.J., Z. Tian and H.Z. Huang, 2007. An efficient method for reliability evaluation of multistate networks given all minimal path vectors. *IIE Trans.*, 39: 811-817. DOI: 10.1080/07408170601013653

Acronyms

- DMPs Disjoint minimal paths.
 SFN Stochastic-flow network.
 QPP Quickest path problem.

Notations

- $G(A, N, M)$ SFN with a set of nodes N , a set of arcs $A = \{a_i | 1 \leq i \leq n\}$ and $M = \{M^1, M^2, \dots, M^m\}$ with M^i (an integer) being the maximum capacity of each arc a_i .
- X Capacity vector; $X = (x_1, x_2, \dots, x_n)$.
- MPs Minimal paths.
- MP_j A minimal path no. j ; $j = 1, 2, \dots, \omega$.
- TP_j The transmission time of path P_j .
- CP_j The maximum capacity of P_j .
- d The given demand, units of data to be transmitted.
- T The time limit, units of time.
- C_s The required system capacity, given in units of cost.
- l_i The lead time of arc a_i .
- R_{d,T,C_s} System reliability for the given demand d under T and C_s .

Nomenclature

- $X \leq Y$ $X = (x_1, x_2, \dots, x_n) \leq Y = (y_1, y_2, \dots, y_n)$, if $x_i \leq y_i$ for $i = 1, 2, \dots, n$
- $X < Y$ $X \leq Y$, when $x_i < y_i$ for at least one i

Lower Vector $X \in \Gamma$ is a lower vector if there is no $Y \in \Gamma$ such that $Y < X$.

Assumptions

1. The capacity of each arc a_i is an integer-valued random variable, which takes values $0 < 1 < 2 < \dots < M$ according to a given distribution.
2. Each node is perfectly reliable.
3. The capacities of different components are statistically independent.