P. DOULLIEZ
E. JAMOULLE

Transportation networks with random arc capacities


<http://www.numdam.org/item?id=RO_1972__6_3_45_0>
TRANSPORTATION NETWORKS WITH RANDOM ARC CAPACITIES

by P. DOULLIEZ (1) and E. JAMOULLE (2)

Summary. — The arc capacities of a transportation network are assumed to be independent discrete random variables and the flow requirements at the sink nodes are known values. This paper presents an efficient method for finding the probability that all flow requirements be satisfied and the probability that a given arc be found in a minimal cut. The method is based upon a decomposition principle which allows the transformation of the network capacity state space into non-overlapping subsets. The expected amount of unsupplied flow is also found. The computational efficiency and domain of application of the method are discussed.

1. INTRODUCTION

The nodes of a transportation network are joined by arcs and the capacity of an arc is an upper bound to the flow that may pass over it. The flow must be sent through the network from « supply » nodes to « demand » nodes. The largest amount of flow that can be sent from a supply node is considered as the capacity of a fictitious arc joining the supply node and a common fictitious supply node, denoted as $S$.

In this paper, the amounts of flow required at the different demand nodes $i$ are known values $d_i$.

It is assumed that the arc capacities are independent discrete random variables.

Hence, the probability that all the flow requirements be satisfied is well defined and will be computed by an efficient method. The probability that a given arc be found in a minimal cut will also be computed (3).

(1) Société de traction et d'électricité, 1040 Bruxelles, chargé de Cours à l'Université de Louvain et aux Facultés Universitaires de Namur.
(2) Société de traction et d'électricité, 1040 Bruxelles.
(3) Since the network has several sink nodes the definition of a minimal cut is here slightly different from the definition that can be found in [3]. We define a minimal cut as a set of arcs separating the common supply node $S$ from one or several demand nodes $i$ such that no supplementary flow can go from $S$ to one of these demand nodes $i$.

Revue Française d'Automatique, Informatique et Recherche Opérationnelle n° nov. 1972.
The method is based upon a decomposition principle which allows the transformation of the network capacity state space into non-overlapping subsets.

In section 2, the decomposition principle is presented and is applied to our specific problem. It is shown in section 3 how the expected amount of unsupplied flow and the probability that a given arc be found in a minimal cut can be computed. Computational efficiency and domain of application are discussed in section 4 and 5. In section 5, a numerical example is given.

2. THE DECOMPOSITION PRINCIPLE

2.1. Notations and Definitions

Let $H_j$ be a discrete random variable ($j = 1 \ldots m$). The $m$ random variables $H_j$ are assumed to be independent. A random variable $H_j$ assumes positive values $h_{j1}, h_{j2}, \ldots, h_{jk_j}$ with probabilities $p_{j1}, p_{j2}, \ldots, p_{jk_j}$ respectively. A point $x$ of the state space can be defined as an $m$-tuple of values $x = (h_{1v_1}, h_{2v_2}, \ldots, h_{mv_m})$ where $v_j$ is a numerical index for $j$ going from 1 to $k_j$. For notational convenience, the index $v_j$ will also be used to designate the value $h_{jv_j}$ itself so that a state point $x$ will be denoted as $x = (v_1, v_2, \ldots, v_m)$. The entire state space is denoted by $X$. The state points $x = (1,1,\ldots,1)$ and $x = (k_1, k_2, \ldots, k_m)$ are called «limiting state points» for $X$.

In our specific problem, $H_j$ is a capacity variable for arc $j$ and $h_{j1}, h_{j2}, \ldots, h_{jk_j}$ are capacity values for arc $j$.

For simplicity, we assume that the values $h_{jv_j}$ are such that

$$h_{j1} < h_{j2} < \ldots < h_{jk_j} \quad (j = 1 \ldots m).$$

A state point $x = (v_1, v_2, \ldots, v_m)$ is called a network state.

Let $Pr[X = x]$ be the probability associated with a point $x \in X$.

We have:

$$Pr[X = x] = p_{1v_1} \cdot p_{2v_2} \cdots p_{mv_m}$$

$$\sum_{v_j = 1}^{k_j} p_{jv_j} = 1 \quad (j = 1 \ldots m)$$

$$\Sigma_{\text{over } x} Pr[X = x] = Pr[X] = 1$$

Revue Française d'Automatique, Informatique et Recherche Opérationnelle
2.2. Statement of the problem

Suppose that any state point \( x = (v_1 \ldots v_m) \) of the state space \( X \) can be recognized as either acceptable or not acceptable. Let \( A^0 \) be the set of acceptable points and \( B^0 \) the set of non-acceptable points. The problem is to compute in an efficient way the probabilities \( \Pr[A^0] \) and \( \Pr[B^0] \). We have:

\[
X = A^0 \cup B^0
\]

\[
\Pr[X] = \Pr[A^0] + \Pr[B^0] = 1
\]

In our specific problem, a network state \( x = (v_1 \ldots v_m) \) is acceptable if the capacities \( v_j \) are such that all demands \( d_i \) can be satisfied. The probability \( \Pr[A^0] \) is the probability that all demands be satisfied and \( \Pr[B^0] \) is the probability that at least one demand be not entirely satisfied.

The state space \( X \) has \( \prod_{j=1}^{m} k_j \) elements, so it is hopeless to solve the problem by considering each point of the state space one after the other. Dealing with subsets in which all state points are acceptable or subsets in which all state points are not acceptable, it can be seen in [1] p. 44 that the problem is still quite complex when some subsets may overlap each other. Therefore, we now present a method for constructing sequences of non-overlapping subsets in which all state points are acceptable and non-overlapping subsets in which all state points are not acceptable.

2.3. Classification rule

Suppose that a state point \( x = (v_1 \ldots v_m) \) is acceptable whenever each \( v_j \) is within the range \( [v^0_j, k_j] \) where \( v^0_j \) is a known critical value for the random event \( H_j \). Thus, \( x = (v_1 \ldots v_m) \) is an acceptable state point if

\[
v^0_j \leq v_j \leq k_j \quad (j = 1 \ldots m).
\]

Suppose also that a state point \( x = (v_1 \ldots v_j \ldots v_m) \) is not acceptable whenever at least one \( v_j \) is within the range \( [1, v^*_j] \) where \( v^*_j \) is a known critical value. Therefore, \( x = (v_1 \ldots v_j \ldots v_m) \) is a non-acceptable state point if \( 1 \leq v_j < v^*_j \) for at least one \( j \).

If the classification rule holds, we can say that a state point \( x = (v_1 \ldots v_j \ldots v_m) \) is « unspecified » whenever \( v_j \geq v^*_j \) for all \( j \) and at least one \( v_j \) is within the range \( [v^*_j, v^0_j] \), i.e., \( v^*_j \leq v_j < v^0_j \).

It is explained in section 3 how the critical capacity values \( v^0_j \) and \( v^*_j \) are obtained in our specific problem.

n° novembre 1972, V-3.
2.4. Non overlapping subsets of $X$

A state point is obtained by assigning numerical values to the indices $v_j$.

In figure 1, a state point $x = (v_1, v_2, ..., v_m)$ can be represented by a broken line through the $v_j$ table ($j = 1, ..., m; m = 5$) and regions $V_A, V_B, V_C$ are defined by the critical values $v_j^0$ and $v_j^*$ as follows:

- $v_j \in V_A$ if $v_j^0 \leq v_j \leq k_j$
- $v_j \in V_B$ if $1 \leq v_j < v_j^*$
- $v_j \in V_C$ if $v_j^* \leq v_j < v_j^0$

\[\begin{array}{c|c|c|c|c|c}
& v_1 = 1 & v_2 = 1 & v_3 = 1 & v_4 = 1 & v_5 = 1 \\
\hline
V_A & \text{region} & & & & \\
V_B & & \text{region} & & & \\
V_C & & & \text{region} & & \\
\hline
v_1 = k_1 & v_2 = k_2 & v_3 = k_3 & v_4 = k_4 & v_5 = k_5
\end{array}\]

Figure 1

The $v_j$ index table ($m = 5$)

Classifying the state points $x = (v_1, v_2, ..., v_j, ..., v_m)$ with respect to the critical values $v_j^0$ and $v_j^*$, it can be seen easily that any state point necessarily belongs to one of the following classes:

- $x$ has its values $v_j$ entirely in $V_A$. Then we say that $x \in A$ where $A$ is a set of acceptable state points ($A \subseteq A^0$);
— $x$ has at least one $v_j$ in $V_B$. Then we say that $x \in B$ where $B$ is a set of non-acceptable state points ($B \subseteq B^0$);

— $x$ has at least one $v_j$ in $V_C$ and no value in $V_B$. Then we say that $x \in C$ where $C$ is a set of unspecified state points.

We have:

$$A \cup B \cup C = X$$

$$A \cap B = \emptyset \quad B \cap C = \emptyset \quad A \cap C = \emptyset$$

![Figure 2](image)

The state space (overlapping subsets)

The sets $B$ and $C$ can be defined as follows:

$$B = \bigcup_{j=1}^{m} B_j$$

$$C = \bigcup_{j=1}^{m} C_j$$
where \( x = (v_1 \ldots v_m) \in B_t \) if \( v_i < v^*_i \)
and \( x = (v_1 \ldots v_m) \in C_t \) if \( v_i < v^0_i \) and if \( v_k \geq v^*_k \) (\( k = 1 \ldots m \)).

The subsets \( B_j \) and \( C_j \) are represented in figure 2. Any two sets \( B_j \) (or \( C_j \)) may overlap each other. There is a one-to-one correspondance between a state \( x \) (represented as a line through the table in figure 1) and a point of the set in figure 2 \((m = 5)\).

Since adding probabilities associated with overlapping subsets \( B_j \) and overlapping subsets \( C_j \) is meaningless, let us transform the subsets \( B_j \) and \( C_j \) into \( B'_j \) and \( C'_j \) as represented in figure 3 such that for any \( j \) and \( k \) \((j \neq k)\) we have:

\[
B'_j \cap B'_k = \emptyset \quad \text{and} \quad C'_j \cap C'_k = \emptyset
\]

*Revue Française d'Automatique, Informatique et Recherche Opérationnelle*
The transformation of subsets \( B_j \) \((C_j)\) into subsets \( B'_j(C'_j)\) can be performed sequentially as follows:

\[
\begin{align*}
B'_1 &= B_1 & C'_1 &= C_1 \\
B'_2 &= B_2 - B'_1 \cap B_2 & C'_2 &= C_2 - C'_1 \cap C_2 \\
B'_j &= B_j - \sum_{r<j} (B'_r \cap B_j) & C'_j &= C_j - \sum_{r<j} (C'_r \cap C_j) \\
B'_m &= B_m - \sum_{r<m} (B'_r \cap B_m) & C'_m &= C_m - \sum_{r<m} (C'_r \cap C_m)
\end{align*}
\]

This can be easily translated in terms of \( v_r \), \( v_l \) and cardinal value of index \( l \) for subsets \( B'_l \) and \( C'_l \) \((l = 1 \ldots m)\):

\[
x = (v_1 \ldots v_m) \in B'_l \text{ if } v_l < v^*_l \text{ (or } v_e \leq v^*_e - 1 \text{) and } v_r \geq v^*_r \text{ for any } r < l
\]

\[
x = (v_1 \ldots v_m) \in C'_j \text{ if } v^*_j \leq v_l < v^*_l, v_r \geq v^*_r \text{ for any } r < l \text{ and } v_r \geq v^*_r \text{ for any } r > l
\]

Thus, overlapping subsets \( B_j \) and \( C_j \) are transformed into non-overlapping subsets \( B'_j \) and \( C'_j \) by taking the increasing order of index \( j \) into account. It is easily seen that \( B'_j \cap B'_k = \emptyset \) and \( C'_j \cap C'_k = \emptyset \) for \( j \neq k \). Also,

\[
B'_j = \emptyset \text{ if } v^*_j = 1 \text{ and } C'_j = \emptyset \text{ if } v^*_j = v^*_j.
\]

### 2.5. Probability Computations

If all sets \( A \), \( B'_j \) and \( C'_j \) are exhaustive and non-overlapping subsets of \( X \), we have:

\[
\Pr [X] = \Pr [A] + \sum_{j=1}^{m} \Pr [B'_j] + \sum_{j=1}^{m} \Pr [C'_j] \quad (1)
\]

It is easy to compute each term of the right-hand side in (1) since limiting state points are known for each subset, as indicated in figure 4.

If we define for each \( j \) \((j = 1 \ldots m)\)

\[
p_j = \sum_{v_j = v^*_j}^{v_j} p_{jv_j} \quad ; \quad q_j = \sum_{v_j = v^*_j}^{v^*_j - 1} p_{jv_j},
\]

\[
s_j = \sum_{v_j = 1}^{v^*_j - 1} p_{jv_j}
\]
we have

\[
\Pr [X] = \prod_{j=1}^{m} (p_j + q_j + s_j)
\]  \hspace{1cm} (2)

\[
\Pr [A] = \prod_{j=1}^{m} p_j
\]  \hspace{1cm} (3)

*Revue Française d'Automatique, Informatique et Recherche Opérationnelle*
\[ \Pr[C_j] = q_j \cdot \prod_{r=1}^{j-1} p_r \cdot \prod_{r=j+1}^{m} (p_r + q_r) \quad j = 1 \ldots m \] (4)

\[ \Pr[B_j] = s_j \cdot \prod_{r=1}^{j-1} (p_r + q_r) \cdot \prod_{r=j+1}^{m} (p_r + q_r + s_r) \quad j = 1 \ldots m \] (5)

Notice that \( q_j = 0 \) if \( v_j^0 = v_j^* \) and \( s_j = 0 \) if \( v_j^* = 1 \).

In the last two equations, ignore the product \( \prod_{r=a}^{b} \) whenever \( b < a \). From equations (2) to (5), it can be shown algebraically that (1) holds.

### 2.6. The decomposition principle

If values \( v_j^0 \) and \( v_j^* \) can be known for the initial state space \( X \), non-overlapping subsets \( A, B_j' \) and \( C_j' \) can be constructed.

Any \( x \in A \) is an acceptable state point and any \( x \in B_j' \) \( (j = 1 \ldots m) \) is a non-acceptable state point. Any \( x \in C_j' \) \( (j = 1 \ldots m) \) is an unspecified state point.

Therefore, the associated probability \( \Pr[A] \) is a part of the sought value \( \Pr[A^0] \) and \( \sum_{j=1}^{m} \Pr[B_j'] \) is a part of the sought value \( \Pr[B^0] \).

Now, each non-empty subset \( C_j' \) may be considered as a new initial state space with modified but known limiting state points. In each \( C_j' \), new critical values \( v_j^0 \) and \( v_j^* \) may be determined, as they have been determined in the initial state space \( X \). Thus, each \( C_j' \) may be again decomposed into new non-overlapping subsets \( B_j'' \) that contain only non-acceptable points, a set \( A \) of acceptable points and new subsets of unspecified points. The probabilities associated with new sets \( A \) and \( B_j'' \) can be computed as in (3) and (5) and are added to the former values and so on... until no subset with unspecified points can be generated and all such subsets have been considered. When this occurs, the probabilities \( \Pr[A^0] \) and \( \Pr[B^0] \) are obtained. The set \( A^0 \) and \( B^0 \) are respectively the union of all sets \( A \) and sets \( B_j' \) that have been constructed.

The procedure is finite. For an initial non-empty state space \( X \), we suppose that \( A^0 \) is non-empty otherwise the problem is solved with \( \Pr[A^0] = 0 \). After decomposing \( X \) into \( A, B_j' \) and \( C_j' \), we have \( A \subseteq X \) but if \( A = X \), \( \Pr[A^0] = 1 \) and the problem is solved. Thus, \( A \subseteq X \) and also each \( B_j' \subseteq X \) and \( C_j' \subseteq X \).

Similarly, at any stage of the procedure, each subset of unspecified state points is strictly contained within the set from which it has been generated.

---

(1) What is here called the « Decomposition Principle » is by no means related to the decomposition principle introduced by Dantzig and Wolfe for solving large-scale linear programs.

\( \text{n\° novembre 1972, V-3.} \)
Consequently, the procedure is finite.

The procedure could be summarized as follows:

— Step 1: Start with initial set $X$. Let $Pr[A^0] = Pr[B^0] = 0$.

— Step 2: Given the limiting state points of $X$, compute the critical values $v_j^0$ and $v_j^*$. The sets $A_j, B_j, C_j$ can then be defined with their own limiting state points. Compute $Pr[A]$ and $Pr[B_j]$ for each $j$.

Let $Pr[A^0] = Pr[A^0] + Pr[A]$ and $Pr[B^0] = Pr[B^0] + \sum_{j=1}^{m} Pr[B_j]$.

— Step 3: Choose a set $C_j$ that has not yet been considered. If no such set exists, go to step 4. Otherwise, let $X = C_j$ and go to step 2.

— Step 4: $Pr[A^0]$ and $Pr[B^0]$ are the sought values. We necessarily have $Pr[A^0] + Pr[B^0] = 1$. END

3. TRANSPORTATION NETWORK WITH RANDOM ARC CAPACITIES

3.1. Critical capacity values

In this section, a method for finding critical values $v_j^0$ and $v_j^*$ is presented for our specific network problem.

They will be found with respect to initial network state set $X$ with limiting state points $x = (1, 1 ... 1)$ and $x = (k_1 ... k_m)$ but the procedure would be the same for any set of network states with known limiting state points. The probability $Pr[A]$ and probabilities $Pr[B_j]$ can be computed once the critical capacity values $v_j^0$ and $v_j^*$ have been determined for each random capacity $H_j$.

Let us join each demand node $i$ to a common fictitious demand node $T$ by way of an arc with capacity $d_i$. A flow $D = \sum_{i} d_i$ is sent from the common source node $S$ to $T$ and the flow $f_j$ in any arc $j$ cannot be higher than $k_j$.

The values $v_j^0$ are obtained as follows:

Let $v_j^0$ be the smallest capacity value that can be assumed by the discrete random variable $H_j$ and which is not less than $f_j$. The network state $x = (v_1^0 ... v_j^0 ... v_m^0)$ is obviously acceptable. The set $A$ is a set of network states $x = (v_1 ... v_j ... v_m)$ with $v_j^0 \leq v_j \leq k_j$ for any $j$. Therefore, any $x \in A$ is acceptable since it has been assumed in the previous section that capacity values are increasing with their indices.

The $j^{th}$ value $v_j^*$ is obtained as follows:

The set $B_j$ is a set of non-acceptable network states $x = (v_1 ... v_j ... v_m)$ with $1 \leq v_j < v_j^*$ for arc $j$. 

Revue Française d'Automatique, Informatique et Recherche Opérationnelle
Given the limiting capacity values $k_r (r \neq j)$ and the existing flow $D$ in the network, find the maximal flow $F_j$ that can still go from the origin to the end of arc $j$ without using arc $j$. The value $F_j$ can be obtained by Ford-Fulkerson's method [3] and corresponds to the largest feasible decrease in flow for arc $j$. If $F_j < f_j$, arc $j$ is in a minimal cut separating node $S$ from node $T$ and $v^*_j$ is the smallest capacity value which is not less than $(f_j - F_j)$. Obviously, any $x \in B_j$ is not able to satisfy all demands and is non-acceptable. If $F_j \geq f_j$, then $v^*_j = 1$ and $B_j' = \emptyset$ since the entire range of capacity values for arc $j$ is feasible for values $d_j$.

3.2. Presence of an arc in a minimal cut

We are now interested in finding the probability $Pr [B^0 | r]$ that arc $r$ be found in a minimal cut ($r = 1 \ldots m$). Since any non-acceptable network state has a minimal cut associated with it, only those with a minimal cut that contains arc $r$ must be retained and their associated probabilities must be summed up.

For any set of non-acceptable network states $B_j'$ that has been generated by the procedure explained in section 2, let us make the following decomposition:

The limiting state points of a set $B_j'$ are known and let us designate them as $x = (g_1 \ldots g_m)$ and $x = (k_1 \ldots k_m)$ with $g_j \leq k_j$ for all $j$. Let $M$ be a minimal cut associated with a network state $x$ that is defined by the highest capacity values, i.e., $x = (k_1 \ldots k_m)$. The value of the cut $M$ is necessarily lower than the flow $D$ which must go from $S$ to $T$ but cut $M$ is not necessarily a minimal cut for network states with capacity values $v_j$ with $g_j \leq v_j \leq k_j$. The decomposition principle must now be applied to the set $B_j'$ with a different criterion, i.e. $B_j'$ must be divided into (non-overlapping) subsets of network states with identical minimal cut.

Let $X = B_j'$. The set $X$ can be decomposed into a subset $A$ and new subsets $B_j' (j = 1 \ldots m)$ when critical capacity values $(v_1 \ldots v_j^0 \ldots v_m^0)$ are determined, we have:

$$x = (v_1 \ldots v_m) \in A \text{ if } v_j \geq v_j^0 \text{ for all } j$$

$$x = (v_1 \ldots v_m) \in B_j' \text{ if } v_j < v_j^0 \text{ for at least one } j.$$

The critical values $v_j^0$ are obtained as follows:

For any arc $j \notin M$, let $v_j^0$ be the smallest capacity value which is not less than the flow value in arc $j$. For any arc $j \in M$, let $v_j^0 = g_j$.

Obviously, any network state $x \in A$ has the minimal cut $M$ associated with it. If arc $r$ is in cut $M$, the value $Pr [A]$ is a part of the sought value $Pr [B^0 \mid r]$ and is to be added to the part already computed.

n° novembre 1972, V-3.
The network states $x$ in a new set $B_j^*$ do not necessarily have an identical minimal cut. Therefore, each new set $B_j^*$ is to be decomposed in the same manner as the initial ones. The value $\Pr[B^0 \mid r]$ is obtained when no subset remains for decomposition.

3.3. Existence of a minimal cut

It is interesting to notice that probability $\Pr[B^0 \mid L]$ that a cut $L$ be minimal can be found similarly by summing up all probabilities associated with sets in which the network states have $L$ as minimal cut. Thus the problem that has been formulated in [4] by Frank and Hakimi can be solved by our approach.

3.4. Expected value of unsupplied flow

Let $A$ be a set of non-acceptable network states with identical minimal cut and with known limiting state points. It has been shown above how an exhaustive sequence of sets $A$ can be generated.

Let the limiting state points of $A$ be $x = (g_1 \ldots g_j \ldots g_m)$ and $x = (k_1 \ldots k_j \ldots k_m)$ with $k_j \leq g_j$ for all $j$.

Let $D$ be the total demand and $G(x)$ be the value of the minimal cut for state $x \in A$.

Each network state $x \in A$ gives rise to an unsupplied flow $(D - G(x))$ with a probability $\Pr[A = x] = p_x$.

The expected value $\mu[A]$ of the unsupplied flow associated with $A$ is:

$$\mu[A] = \sum_{x \in A} p_x (D - G(x)) = \Pr[A] \cdot D - \sum_{x \in A} p_x G(x)$$

The term $\sum_{x \in A} p_x G(x)$ is the expected value of minimal cut $M$ within set $A$.

Let

$$q_j = \sum_{v_j = g_j}^{k_j} p_{j,v_j}$$
$$h_j = \sum_{v_j = g_j}^{k_j} h_{j,v_j} \cdot p_{j,v_j}$$

$M$ be the minimal cut for any $x \in A$.

From the fact that the expected value of a sum of independent random variables is the sum of the expected values of these variables, we have:

$$\sum_{x \in A} p_x G(x) = \prod_{j \in M} q_j \left( \sum_{j \in M} h_j \cdot \prod_{r \in M, r \neq j} q_r \right)$$
$$= \sum_{j \in M} h_j / q_j \cdot \Pr[A]$$
Therefore, we have:

$$\mu[A] = \Pr[A] \cdot \left( D - \sum_{j \in M} \bar{h}_j / q_j \right)$$  \hspace{1cm} (1)$$

Computing the right-hand side of (1) is straightforward. Summing up the values in (1) for all generated sets $A$ gives the expected value of the unsupplied flow.

Similarly, summing up the values in (1) for generated sets $A$ for which the minimal cut contains arc $r$ gives the expected value of the flow which is unsupplied when arc $r$ is present in a minimal cut.

Finally, summing up the values in (1) for generated sets $A$ for which cut $M$ is minimal for any $x \in A$ gives the expected value of the flow which is unsupplied when cut $M$ is minimal.

4. CONCLUDING REMARKS

The decomposition principle presented in this paper is general in the sense that it could be applied in any other system in which a point of the state space is defined when several random events occur simultaneously and when a point of the state space can be classified according to a given criterion.

The problem of finding the probability of meeting the demands at the nodes of a network has been treated up to now by making a large number of successive random trials (Monte-Carlo approach). The method presented in this paper is able to find an exact answer to the problem with a small computational effort. Anyway, Monte-Carlo approach is not appropriate for estimating properties of a set of state points which is reached very rarely even after a great number of random trials. For some networks that have been treated by our approach, the probability associated with all non acceptable state points was not higher than 0.001. A main reason for efficiency is that a sequence of non-overlapping sets of network states are considered instead of a sequence of single network states as in the Monte-Carlo approach.

When the demands at the demand nodes are increasing with time, the method can be easily adapted for finding the largest value of time up to which all demands can be satisfied with a given probability level.

Also, a best location for an investment on a network can be found since the responsibility of each arc for non satisfying the demands is evaluated with respect to the entire network.

The present paper can be considered as a probabilistic extension of reference [2]. In [2], the required probability level with which all demands must be satisfied has been taken into account only in an empirical way — i.e. — all demands must be satisfied even if any one arc assumes a given lower capacity.
Several numerical examples have been treated on an IBM 370-155 by a program written in FORTRAN IV. An example with about 20 random arcs usually needs less than one second of computing time.

5. NUMERICAL EXAMPLE

Consider the network in figure 5. In the same figure, the flow requirements at the demand nodes are indicated. Table 1 gives the capacity values that can be assumed by the arcs of the network and their associated probabilities.

In figure 5, two data are associated with each arc. The first one is the probability that the arc be found in any minimal cut and the second is the expected value of the flow which is unsupplied when the arc is present in a minimal cut.

In this example, the probability that at least one demand be not entirely satisfied is found to be 0.1418 and the expected value of the unsupplied flow is 18.93.

Looking at figure 5, it can be concluded that arc S-5 must be reinforced in priority since it is present in a minimal cut for almost each non-acceptable

*Revue Française d'Automatique, Informatique et Recherche Opérationnelle*
network state \((0.134 \approx 0.1418)\). Also, the expected value of the flow which is unsupplied when the arc \(S-5\) is present in a minimal cut is almost equal to the unconditional expected value of unsupplied flow \((18.20 \approx 18.93)\).

### Table 1

<table>
<thead>
<tr>
<th>Arc</th>
<th>Capacity values and associated probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S-1)</td>
<td>800 (1.0)</td>
</tr>
<tr>
<td>(S-2)</td>
<td>0 (0.0035)</td>
</tr>
<tr>
<td>(S-3)</td>
<td>0 (0.04)</td>
</tr>
<tr>
<td>(S-4)</td>
<td>0 (0.05)</td>
</tr>
<tr>
<td>(S-5)</td>
<td>0 (0.05)</td>
</tr>
<tr>
<td>(1-6)</td>
<td>0 (0.0053)</td>
</tr>
<tr>
<td>(1-8)</td>
<td>0 (0.0047)</td>
</tr>
<tr>
<td>(1-9)</td>
<td>0 (0.0047)</td>
</tr>
<tr>
<td>(2-3)</td>
<td>24 (1.0)</td>
</tr>
<tr>
<td>(3-6)</td>
<td>0 (0.0225)</td>
</tr>
<tr>
<td>(3-7)</td>
<td>0 (0.0040)</td>
</tr>
<tr>
<td>(3-10)</td>
<td>0 (0.0001)</td>
</tr>
<tr>
<td>(4-7)</td>
<td>0 (0.0026)</td>
</tr>
<tr>
<td>(4-10)</td>
<td>0 (0.0001)</td>
</tr>
<tr>
<td>(5-10)</td>
<td>0 (0.0031)</td>
</tr>
<tr>
<td>(5-12)</td>
<td>0 (0.00001)</td>
</tr>
<tr>
<td>(6-10)</td>
<td>0 (0.0004)</td>
</tr>
<tr>
<td>(8-6)</td>
<td>(\infty) (1.0)</td>
</tr>
<tr>
<td>(8-11)</td>
<td>(\infty) (1.0)</td>
</tr>
<tr>
<td>(9-6)</td>
<td>(\infty) (1.0)</td>
</tr>
<tr>
<td>(9-11)</td>
<td>(\infty) (1.0)</td>
</tr>
<tr>
<td>(10-12)</td>
<td>0 (0.00001)</td>
</tr>
</tbody>
</table>

### REFERENCES


\(\text{n° novembre 1972, V-3.}\)