

Evacuation transportation planning: A method of identify optimal independent routes

Vania B.G. Campos, Paulo A.L. da Silva & Paulo O.B. Netto
*Military Institute of Engineering, De/2, Praca General Tiburcio 80,
Praia Vermelha, 22290-270, Rio de Janeiro, Brasil*
*Email: vania@taurus.ime.eb.br, paulo@taurus.ime.eb.br,
boaventu@pep.coppe.ufrj.br*

Abstract

This paper presents a method which identifies k-optimal independent paths for vehicle flow allocation in emergency transportation planning. The algorithm identifies such path sets in a way that a greater number of vehicles can be sent in a minimum time outside a region under menace of some catastrophic event.

1 Introduction

The Transportation Evacuation Planning is an important part of Emergency Plans, mainly in regions near Nuclear Power Plants. Techniques and models have been developed to simulate the evacuation of people living near this type of plant and also in other areas such as chemical industries. These techniques have been employed in several countries with the main objective of identifying the problems that can happen in the moment of evacuation; these problems can be the vehicular congestion and accidents that contribute to increasing the time of evacuation and causing injury to people. Examining the results of the simulation techniques, the optimal evacuation routes from the target area to the population gathering points or shelters can be determined.

In this paper it is presented an algorithm which identifies k-optimal independent paths for vehicle flow allocation in emergency transportation planning without using simulation technique. This type of routes has no intersection points (they are called disjoint paths), and so, the use of them minimizes the problem of accidents and permits a continuous traffic flow.

2 Characteristics of The Method

According Graph theory (Boaventura¹, Bollobás²) it is known that between a pair of nodes of a network there is a limited number (k) of independent paths (called disjoint paths); however it is possible to exist several sets (of k elements) of these paths in a same network.

The developed method allows to identify the set of the best independent routes. An index defined by the ration C/T (capacity / travel time in the route) is determined for each route, and the best route set is that which presents the largest sum of these indexes. The reason to choose this index is because comparatively a path is better than other if it has larger capacity or a smaller travel time. If two paths have the same capacity, the path which has the smaller travel time should be chosen (its index is larger) and if they have the same time the one which has the larger capacity is better.

In summary, considering the possibility of existing independent k-paths between a pair of source and destination node in a network, the method determines a set of routes with the objective of maximize the summation of the ration of each path, that is,

$$\text{Max} \sum_{p=1}^k C_p / T_p \quad p = 1 \dots k \quad (1)$$

To solve this problem it was developed the following algorithm .

3 Algorithm for Finding k-Independent Routes

The algorithm proposed is an heuristic algorithm. The basic idea of it is to use interactively an algorithm for finding the k-shortest paths in the network, with the smaller travel time, associating to each one an index C/T (capacity/time) that will facilitate to evaluate each route comparatively with another. The routes found at each iteration can present coincident nodes, which should be removed from the network, to verify the existence of others non coincident routes in another iteration.

The objective of the method is to identify k-routes which can drain away the traffic flow from the origin to the destination node in the smaller time.

To arrive to the solution, to each iteration it can remove arcs (links), considered “bottlenecks”(arcs of minimum capacity in the path), or can remove coincident nodes. The exclusion of the bottleneck arcs is made to obtain an improvement in the capacity of routes just found and the exclusion of the coincident nodes is to find paths not coincident with the resultant paths at others interactions. Through several iterations, different routes are found with theirs respective C/T indexes. At the end of the process, the set of the best independent k-routes are chosen.

The proposed algorithm is an iterative process which excludes arcs or nodes of the network at each iteration, which the nodes and the arcs removed from two different networks, obtained from the original network. In a represented network, called RG, the bottleneck arcs are removed and, in the other, called RN, the coincident nodes.

To begin the search, it is necessary to define the number of paths to identify; because the time of processing of the algorithm is function of the number of paths to examine in each iteration. Thus, to define a maximum limit for k means to optimize the time of processing.

The value of k can be defined using a method proposed by Campos³, which determine the maximum number of all possible disjoint paths (k) between a pair of nodes in an oriented network.

For finding k-shortest paths in a network, an existent algorithm is used. It exists some algorithms (Shier⁴, Yen⁵) in the literature that present similar structure. The peculiarities of the studied problem indicate the use of those algorithms whose results do not determine path with “loops” (paths passing more than once by a same node).

3.1 Structure of the Algorithm

Considering a network R with a source node S and a destination node D the algorithm has the following steps:

Step 1: define the number of minimum paths (k) to be search

Reminding that, as discussed above the number k of paths can be defined by the method proposed by Campos³.

Step 2: determine the k-shortest paths

The determination of the k-shortest path between the source S and the destination D in the network it is made by using an existent algorithm.

For each one of the found paths it is obtained the total time:

$$T_p = \sum_{a \in p} t(a)_p \quad \text{for } p = 1, 2, \dots, k \quad (2)$$

where $t(a)_p$ is the travel time in the arcs that belongs to the path p .

If it is the first iteration, go to step 3; otherwise, two situations can happen:

(1) do not exist the paths between the source and the destination nodes; in this case, if in the previous iteration it were removed arcs (step 4) goes to the step 6, if in the previous iteration it were removed nodes (step 6), go to step 4. If for two consecutive iterations there is not possibilities to find paths, then go to step 7.

(2) there are paths between the source and destination nodes, go to step 3.

Step 3: Identify the bottleneck arcs and define the indexes C_p/T_p

It is verified the maximum flow that can be allocated in each found path, defined by the arc with the smaller capacity (bottleneck arch of the path, (g_p)) in the path .

It is defined the capacity $C_p = \min C(i,j)_p$ and $I_p = C_p/T_p$ (capacity/ travel time index) for each path p , $p=1,...,k$.

Each passage in the step 2 defines an iteration that is concluded in the step 3. The found paths and their respective indexes are stored in a list of rk paths, where r represents the number of iterations. In the first iteration ($r=1$) it is used the original represented network.

If the next iteration is an odd iteration, go to step 4 for removing bottleneck arcs; otherwise, go to step 5 for identifying the coincident nodes.

Step 4: Modify the network removing arcs

Remove the bottleneck arcs identified in the two previous iterations (or one, if it is the second iteration), and return to the step 2. This modification is made in the network RG for a reappraisal of the paths regarding their capacity and the respective indexes I_p .

Be r the current iteration. Then, RG^{r-2} is the network used in this step in the iteration $r-2$, from which it was removed bottleneck arcs. We have:

$$RG^r = RG^{r-2} - \{g^{r-1}_p, g^{r-2}_p\} \quad p = 1, 2 \dots k \quad (3)$$

where g^{r-1}_p and g^{r-2}_p are, respectively, the set of bottleneck arch of the two previous iterations.

With the network RG^r it goes back to step 2.

Step 5: Identify coincident nodes

In this step, it is verified the existence of coincident nodes among the paths found until now. If they exist, verify those with the largest frequency.

Be L the number of paths found until the present iteration, and x_l a node of a path l ($l = 1, 2, \dots, L$). It is defined $|x_l|$ as the number of paths where the node x_l appears.

If $|x_l| \leq (L - (k - 1))$, independent k -routes can be found between result paths. In this case go to step 7; otherwise go to step 6.

Step 6: Modify the network removing nodes

Remove the coincident nodes from the RN-network, or from the initial network R , if this is the first passage for this step, and return to the step 2.

The nodes to be removed are those identified in the previous step. Be XL the group of them and be RN^{r-2} the network modified by removing the nodes in the previous passage by this step.

$$\text{Let's do: } RN^r = RN^{r-2} - \{XL\} \quad (4)$$

and go to step 2.

Step 7: Identification and choice of k -Independent routes sets

In this step, starting from a group L of paths found in r iterations, it is verified the existence of independent paths. If they exist, they are put in sets of independent k -paths. Two situations can happen:

1) there are sets of k paths, in this case the sum of their respective indexes (C_p/T_p) is calculated, being chosen the set with the largest sum. To decide if this is the best solution the following analyses can be made:

(i) if there is no more possibility to find paths in both networks (RG and RN), then the existent solution is the best.

(ii) if there is the possibility to find paths, the iterations continue and we return to step 6.

2) there are no sets with k -elements:

(i) if does not exist possibility of finding paths and exist set of paths with less of k elements, the found solution is that which gives the largest sum of indexes among the found sets; otherwise there is not solution.

(ii) if it still exists possibility of finding paths, go to step 6.

End of the algorithm.

4 Application of The Method

The planning of the evacuation includes the definition of the structure of the network and its characteristics, a forecast of the demand of traffic generated by the population to be evacuated, the definition of

the place or safe places for which the population should go and, consequently, the best routes to arrive to these locals.

When an accident happens in a risk area, the evacuation should be done in stages: initially, it is promoted the evacuation of the closest place of the disaster area, and later on according to a planning, the evacuation of the other areas in agreement with the extension of the disaster.

For a better understanding of the developed method and to show its viability, it is presented an application in suburb of Bangu in Rio de Janeiro city.

The network structure has 46 nodes and 74 arcs. Table 1 presents the travel times on the arcs (minutes) and the flow capacities of them (vehicles/hour).

Table 1 - Characteristics of the Bangu's Network

arc s,d	time (min)	capac. veic/h	arc o,d	time (min)	capac. veic/h	arc s,d	time (min)	capac
1,2	1	3098	15,25	10	1575	30,31	2	1960
1,4	1	2795	16,19	1	3150	31,32	2	1960
1,5	1	2505	16,17	1	1750	31,36	1	1700
2,3	5	1500	17,20	1	3570	32,33	2	1960
3,9	10	1540	18,20	5	1525	32,37	2	3400
4,3	4	1098	18,21	1	2940	33,38	2	3000
4,6	6	2695	19,22	5	3150	33,44	5	2450
5,6	5	2254	20,19	1	1960	34,39	1	1200
5,7	3	2340	20,23	1	3570	35,34	1	3881
6,8	8	2800	21,24	1	2600	35,40	1	2925
7,10	4	2940	21,20	1	1960	36,35	1	3881
8,12	7	2940	22,23	3	1588	37,36	1	3881
8,13	4	1810	22,26	1	3150	37,42	4	3400
8,10	2	1820	23,27	2	2600	38,37	1	3881
9,11	9	1540	23,24	3	1588	38,43	1	3881
9,12	7	2940	24,28	2	2600	39,40	1	1126
10,13	4	1820	25,29	3	1575	40,41	1	2025
10,14	2	2940	26,25	5	1360	41,42	1	2025
11,15	9	1540	26,30	3	3150	41,45	3	2400
12,11	2	1190	27,32	2	2600	42,43	1	2025
12,16	5	3080	28,33	1	2600	42,46	3	3400
13,17	10	2940	28,27	1	1588	43,46	3	2800
14,13	2	1820	29,30	1	1000	44,46	1	2925
14,18	10	2940	29,34	1	1200	45,46	1	2400
15,16	1	1120	30,35	1	2925			

The flow of vehicles to be evacuated has origin on the node 1, central point of an area that is characterized as an area of risk and that should be evacuated immediately in the case of the effective event of the catastrophic. The destination node is the node 46, being considered a safe place for where it should be move the evacuated population.

4.1 - Identification of the Best Independent Routes

Step 1 Define k

The method proposed by Campos³ verified that the number of possible disjoint paths between the node 1 and the node 46 is 3. Then $k=3$.

1st iteration

Step 2 Find the 3-shortest paths in the original network (table 2).

Table 2- The 3- shortest paths

Path order :	path nodes:	total time:
1	1-5-7-10-14-18-21-24-28-33-38-43-46	31
2	1- 5-7-10-14-18-21-24-28-33-44-46	31
3	1-5-7-10-14 -18-21-24-28-33-38-37-42-46	35

Step3: Capacity, index arch and bottleneck arcs

The capacity of the paths 1,2 and 3 are 2340 veic/h gave by the bottleneck arc (5,7) and the indexes: 75,48; 75,48 and 66,86 respectively.

As this is the first iteration, goes to step 4.

Step 4: Modifying the original network R^0 (removing arcs).

Remove from the original network the bottleneck arcs identified in the previous step and define: $RG^1 = R^0 - \{(5,7)\}$

2^a iteration

Step 2 and Step 3 : Finding 3-shortest paths in the network RG^1 (table 3) and Capacity, indexes and bottleneck arcs of the paths.

Table 3 - The 3-shortest path in the network RG^1

Path order	nós do caminho	total time
4	1-5-6-8-10-14-18-21-24-28-33-38-43-46	39
5	1-5-6-8-10-14-18-21-24-28-33-44-46	39
6	1-4-6-8-10-14-18-21-24-28-33-38-43-46	40

The capacity of the paths 4,5 and 6 is 1820 v/h given by the bottleneck arc (8,10) and indexes : 46,67; 46,67 and 45,50 respectively.

As in the previous iteration were removed bottleneck arcs, go to step 5.

Step 5: Identifying coincident nodes.

In all the six paths it is identified the following coincident nodes: 10,14,18,21,24,28,33. Each one has $|x_i| = 6 \forall i$, and as $L = 6$ (6 paths found until now) and $|x_i| > L-(k-1)$, then, goes to step 6.

Step 6: Modifying the original network R^0 (removing nodes)

Removes, from the original network, the coincident nodes identified previously, and defines:

$$RN^1 = R^0 - \{10,14,18,21,24,28,33\}$$

Go to step 2.

3rd iteration

Step 2 and Step3: Find 3-shortest paths in the network RN^1 (table 4) and Capacity, Indexes and bottleneck arc of the paths .

Table 4 - The 3-shortest paths in RN^1

Path Order	Path nodes	total time
7	1-5-6-8-12-16-17- 20-23-27-32-37-42-46	42
8	1-4-3-9-12-16-17- 20-23-27-32-37-42-46	43
9	1-4-6-8-12-16-17- 20-23-27-32-37-42-46	43

The capacity of the paths 7, 8 and 9 is 1750 v/h, 1098 v/h and 1750v/h respectively, given by the bottleneck arcs: (16,17), (4,3) and (16,17) of each one. The index are in path order: 41,67; 25,53 and 40,70.

As in the previous step they were removed coincident nodes, go to step 4.

Step 4: Modifying the network RG^1

To remove of the network RG^1 the arcs bottleneck arcs of the paths obtained in this and in the previous iteration:

$$RG2 = RG1 - \{(4,3), (16,17), (8,10)\}$$

To go for the step 2.

4th iteration

Step2 and Step 3 Find 3-shortest paths in the network RG^2 and the capacity, index and bottleneck arc of the paths.

Table 5 - The 3-shortest paths in RG^2

Path order	Path nodes	total time
10	1-5-6-8-13-17-20-23-24-28-33-38-43-46	42
11	1-5-6-8-13-17-20-23-24-28-33-44-46	42
12	1-5-6-8-13-17-20-23-27-32-33-38-43-46	42

The capacity of the paths 10, 11 and 12 is 1588 v/h, 1588 v/h and 1810v/h respectively given by the bottleneck arcs: (23,24), (23,24) and (8,13) of each one. The index are in path order: 37,81; 37,81 and 43,10.

As in the previous iteration it was removes the bottleneck arcs, then go to step 5.

Step 5 : Identifying the coincident nodes

It is verified that the node 5 appears the largest number of times in the paths found until now ($L = 12$); this node appears in nine paths ($|x_5| = 9 < L - (k - 1)$), it indicates the possibility of existence of independent paths among the found paths. Go to step 7.

Step 7: Identification of K- independent routes

It was only identified sets of two independent paths, and as it is being searched sets of tree independent routes, go back to step 6.

Step 6 : Modifying the Network RN^1

Let's made: $RN^2 = RN^1 - \{5\}$

Go to step 2.

These steps are repeated until the set of tree-independents routes are found, and, in this example, it happened at the 11^{en} iteration. The table 6 shows set of tree- independent routes found.

Table 6 - The 3-optimal independent path identified

Path (set 1)	Path nodes	t. time	index
1	1-5-7-10-14-18-21-24-28-33-38-43-46	31	75.48
9	1-4-6-8-12-16-17- 20-23-27-32-37-42-46	43	40.70
33	1-2-3-9-11-15-25-29-34-39-40-41-45-46	55	20.47

Another set with the paths: 2,9 and 33 has the same sum of index and it is a solution too. The result (set 1) is illustration in the figure 1.

5 Final considerations

The proposed method was tested in another network cases, besides in no-oriented networks, and it was verified its efficacy and usefulness. A program was developed in Pascal language for these tests, even so the program needs to be improved.

The importance in modeling the network should be emphasize for larger success of the results. Moreover the network should contain the main roads of the area, those that present the best physical conditions for the traffic, beyond the definition of the source (origin) and destination nodes in function of the risk area and the shelter places. The origin of the flow can be only one, if there is more than one place (node) it makes necessary to create a fictitious origin that will be linked to the several origins. The same is done in relation to the destination.

The proposed method allows a previous analysis of several scenarios including: the total flow of vehicles in the evacuation, alteration in the orientation of the roads and the location of the shelter places for the population, and it can be associated to a simulation process.

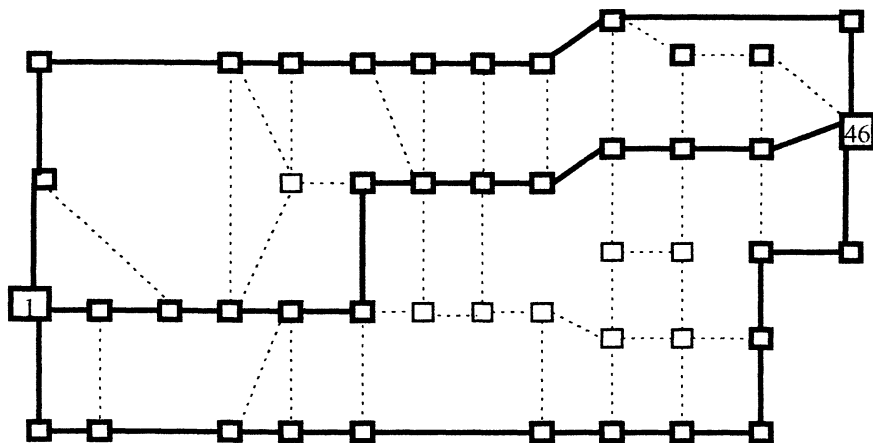


Figure 1 : Routes determined by the method in Bangu's Network

References

- [1] Boaventura Netto, P.O. *Grafos: Teoria, Modelos, Algoritmos*, Editora Edgar Blucher Ltda, São Paulo, 1996.
- [2] Bollobás B., *Extremal Graph Theory*, Academic Press, New York, 1978.
- [3] Campos, V.B.G., *Método de Alocação de Fluxo no Planejamento de Transporte em Situações de Emergência: Definição de Rotas Disjuntas*, Tese de Doutorado, Eng. de Produção COPPE/UFRJ, March, 1997.
- [4] Shier D. R., *Interactive Methods for Determining the K- Shortest Paths in a Network*, *Networks*, No 6, pp 205-229, 1976.
- [5] Yen J.Y. *Finding the K- shortest Loopless Paths in a Network*, *Management Science*, vol.17, No 11, Jul, 1971.