PLANNING AND MANAGEMENT OF MOTORCICLES TOWN SERVICE USING ALL-OR-NOTHING ASSIGNMENT AND FLOYD'S ALGORITHM

PRESENTATION AT NAMP CONFERENCE, LANDMARK, OMU-ARAN
6TH – 9TH NOV., 2018

By

Ngutor Nyor and Mohammed Idrisu ,

Department of Mathematics, Federal University of Technology Minna

ABSTRACT

The paper explored Urban Traffic Planning and Management in Bida town of Niger State using conventional transportation models of All-or-Nothing Traffic Assignment Technique and floyd's Algorithm to find minimum links and associated link flow volumes of commercial motorcycles that ply the routes in the town. The paper presented traffic counts (inflows and outflows) of commercial motorcycles in eight selected Centroids in the town during three peak periods (7am – 8am, 1pm - 2pm and 4pm - 5pm). Minimum paths were identified to ensure good movement/flow of traffic and to reduce cost for the commercial motorcyclists.

Key Word: *Transportation Model, Trip Generation, Trip Distribution, Modal Choice, Assignment Technique, All-or-Nothing, Floyd's, Algorithm.*

INTRODUCTION

Complicated road environments, dense highway traffic networks, and random congestion in urban systems aggravate the difficulties of logistic transportation. Traffic jams result in decreased speed of commercial motorcycles, increase in urban logistic transportation costs, and decrease in the customer service level. Hence, establishing rational urban transportation routes with high efficiency, improving the timeliness of the logistic system, and reducing generalized logistic costs are urgently required. (Hui, 2014).

ALL-OR-NOTHING

The all-or-nothing assignment method involves the concept of traffic distribution, planning, and management. Traffic assignment refers to the process in which existing origin–destination (OD) trips are assigned to various paths of the network according to a specific assignment algorithm to obtain the assignment flow of each OD at each road segment and the total flow of each road segment. Supposing that the impedance of a road segment is a constant (the traveling time is not influenced by the traffic flow of this road segment), all trips of a producing point are assigned; an attracting point is likewise assigned to the shortest path between such points at one time, and no point is assigned to other road segments. Such assignment method is called the all-or-nothing assignment method (Chen, 1012; Calum, 2012; Yan-Bing et al., 2014)

FLOYD'S ALGORITHM

The Floyd's Algorithm also called Floyd–Warshall algorithm is a shortest path algorithm. It is an example of dynamic programming which was published in its currently recognized form by Robert Floyd in 1962. The Floyd–Warshall algorithm compares all possible paths through the graph between each pair of vertices. Floyd's algorithm is more general than Dijkstra's because it determines the shortest route between any two nodes in the network

The algorithm represents an *n-node* network as a square matrix with *n* rows and *n* columns. Entry (*i*,j) of the matrix gives the distance *dij* from node *i* to node j, which is finite if *i* is linked directly to j, and infinite otherwise.

OBJECTIVE OF THE STUDY

The objective of this paper is to identify the minimum link path and volume of trips between given origins and destinations in Bida town using all-or-nothing assignment technique and Floyd's algorithm.

MATERIALS AND METHODS

According to Taha (2010), the idea of Floyd's algorithm is straightforward. Given three nodes *I*, j, and *k* in Figure 1 with the connecting distances shown on the three arcs, it is shorter to reach *j* from *i* passing through *k* if

$$d_{ik} + d_{kj} < d_{ij}$$

In this case, it is optimal to replace the direct route from $i \rightarrow j$ with the indirect route $i \rightarrow k \rightarrow j$. This triple operation exchange is applied systematically to the network using the following steps:

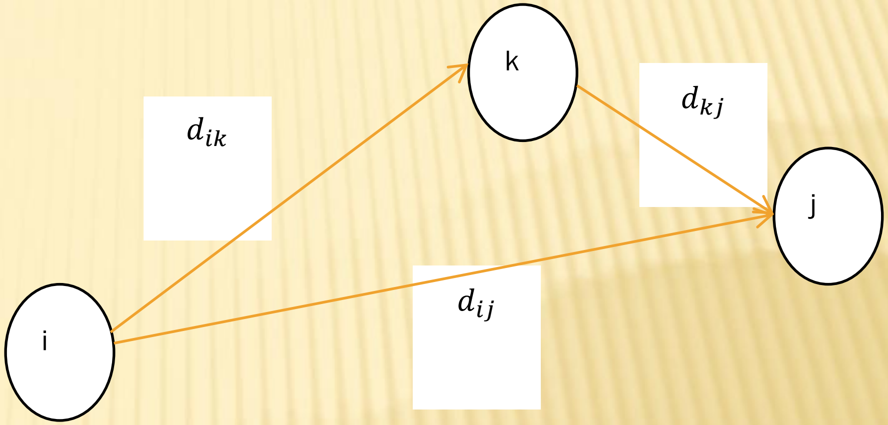

Figure. 1: Floyd's Triple Operation

METHOD CONT…

Step 0: Define the starting distance matrix D_0 and node sequence matrix S_0 as given below. The diagonal elements are marked with (–) to indicate that they are blocked. Set k = 1.

General step ***k:*** Define row *k* and column *k* as *pivot row* and *pivot column.* Apply the *triple operation* to each element d_{ij} in $D_{k-1,}$ for all *i* and *j*. If the condition

$d_{ik} + d_{kj} < d_{ij}$, (i ≠ k, j ≠ k, and I ≠ *j*)

is satisfied, make the following changes:

(a) Create D_k by replacing d_{ij} in D_{k-1} with $d_{ik} + d_{kj}$

(b) Create S_k by replacing S_{ij} in S_{k-1} with k. Set $k = k + 1$. If $k = n + 1$, stop; else repeat step *k*.

D	1	2	…	j	…	n
1	–	d_{12}	…	d_{1j}	…	d_{1n}
2	d_{21}	–	…	d_{2j}	…	d_{2n}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	d_{i1}	d_{i2}	…	–	…	d_{in}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	d_{n1}	d_{n2}	…	d_{nj}	…	–

S	1	2	…	J	…	n
1	–	2	…	J	…	n
2	1	–	…	J	…	n
⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	1	2	…	J	…	n
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	1	2	…	J	…	–

DATA PRESENTATION

The survey on motorcycle (popular called Byke or Okada or Kabukabu) was carried out in Bida, Bida Local Government Area of Niger State. Below is the road network digraph of 8 zones which were considered along with arc weights as the average time (in minutes) it takes a motorcycle at 40km/h to move from an origin to a destination.

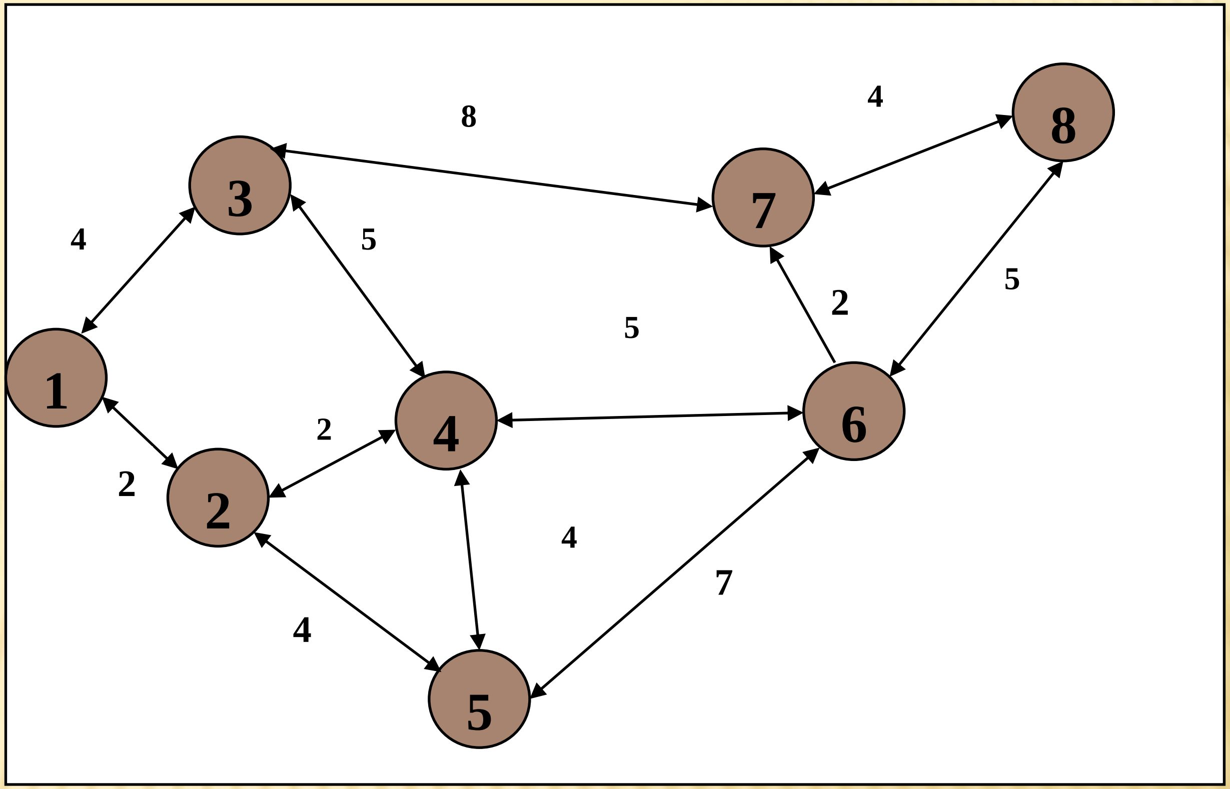

Figure 2: A Digraph of TAZs

Table 1: TAZs Defined

1	CIRICO JUNCTION	2	ESSO JUNCTION
3	BANGAYE	4	LUNCHITA
5	WADATA	6	POLY JUNCTION
7	BANYAGI	8	FEDERAL POLY BIDA

Table 2: Link Array Table

From /To	1	2	3	4	5	6	7	8
1	–	2	4	∞	∞	∞	∞	∞
2	2	–	∞	2	4	∞	∞	∞
3	4	∞	–	5	∞	∞	8	∞
4	∞	2	5	–	4	5	∞	∞
5	∞	4	∞	4	–	7	∞	∞
6	∞	∞	∞	4	7	–	2	5
7	∞	∞	8	∞	∞	2	–	4
8	∞	∞	∞	∞	∞	5	4	–

TABLE 3: TRAFFIC PRODUCTIONS (OUTFLOW) AND ATTRACTION (INFLOW) DURING PEAK PERIODS OF THE TAZS

	7 – 8AM		1 – 2PM		5 – 6PM	
Zone	Production Outflow	Attraction Inflow	Production Outflow	Attraction Inflow	Production Outflow	Attraction Inflow
1	550	700	450	650	500	600
2	670	860	560	800	750	500
3	350	280	300	430	400	550
4	700	950	640	880	850	720
5	550	750	650	600	720	450
6	400	640	300	520	320	550
7	420	400	330	480	360	440
8	580	800	500	740	750	560

Note: This was head count of motorcycles as they were produced from and attracted to zones

PROBLEM FORMULATION
TABLE 4: TRIP GENERATION: AVERAGE TRIP PRODUCTION AND ATTRACTION IN THE TAZS

Zone	Production (Outflow)	Attraction (Inflow)
1	500	650
2	660	720
3	350	420
4	730	850
5	640	600
6	340	570
7	370	440
8	610	700
TOTAL	4200	4950

Observe from table 4 that the total outflow is not equal to total inflow. Thus, we adjust trip attractions to match total attraction equal to total production as shown in the table 5. Usually, the attractions are modified to make their sum equal to that of the production. This is done by applying the following formula to each attraction for each zone:

Adjusted Trip Attraction $(ATA_j) = \frac{\sum_{i=1}^{N} P_i}{\sum_{j=1}^{N} A_j}$ x A_j ; where *P* is Production and *A* is attraction.

TABLE 5: ADJUSTED TRIP INFLOW IN THE TAZS

Zone	Trip Production (Outflow)	Adjusted Trip Attraction (Inflow)
1	500	552
2	660	611
3	350	356
4	730	721
5	640	509
6	340	484
7	370	373
8	610	594
TOTAL	4200	4200

Observe that total trip production equals attraction

TRIP DISTRIBUTION

The number of trips between any two zones is obtained as shown in the table 6. This is done using the formula

$D_{ij} = \frac{P_i \times A_j}{T}$; where D_{ij} is the distance from *i* to *j*, P_i is Production at *i*, A_j is Attraction at *j* and *T* is the Total of productions and Attractions which is the same. For example

$D_{11} = 0$, $D_{12} = \frac{500\ X\ 611}{4200} \approx 73$, $D_{13} = \frac{500\ X\ 356}{4200} \approx 42$, $D_{14} = \frac{500\ X\ 721}{4200} \approx 86$, $D_{15} = \frac{500\ X\ 509}{4200} \approx 61$

Table 6: Trip Distribution Table

Zone	1	2	3	4	5	6	7	8	Obtained (P_i)	Actual (P_i)	Row Factor (F_i)
1	0	73	42	86	61	58	44	71	435	500	1.15
2	87	0	56	113	80	76	59	93	564	660	1.17
3	46	51	0	60	42	40	31	49	320	350	1.09
4	96	106	62	0	88	84	65	103	605	730	1.21
5	84	93	54	110	0	74	57	91	562	640	1.14
6	45	49	29	58	41	0	30	48	301	340	1.13
7	49	54	31	64	45	43	0	52	337	370	1.10
8	80	89	52	105	74	70	54	0	524	610	1.16
Obtained (A_j)	486	515	327	596	432	444	340	508	3647	4200	
Actual (A_j)	552	611	356	721	509	484	373	594	4200		
Column Factor (F_j)	1.14	1.19	1.09	1.21	1.18	1.09	1.10	1.17			

TABLE 7: ADJUSTED TRIP DISTRIBUTION TABLE (FOURTH ITERATION)

Calculations of New trip distributions can now be done by obtaining the column and row balancing factors. This is done iteratively using

$DT^1_{ij} = F^0_j \text{ x} D^0_{ij}$, for example

$D^1_{12} = F^0_2 \text{ x} D^0_{12}$ = 1.19 x 73 ≈ 87; $D^1_{13} = F^0_3 \text{ x} D^0_{13}$ = 1.09 x 42 ≈ 46; $D^1_{14} = F^0_4$ x D^0_{14} = 1.21 x 86 ≈ 104 … $D^1_{87} = F^0_7$ x D^0_{87} = 1.09 x 54 ≈ 59.

Zone	1	2	3	4	5	6	7	8	Obtained	Actual	Row Factor
1	0	87	46	104	71	62	48	82	500	500	1.00
2	100	0	62	141	96	84	65	112	660	660	1.00
3	49	57	0	69	47	41	32	55	350	350	1.00
4	115	134	71	0	110	97	75	128	730	730	1.00
5	95	109	58	133	0	79	61	105	640	640	1.00
6	49	57	30	69	47	0	33	55	340	340	1.00
7	52	61	32	74	50	44	0	57	370	370	1.00
8	92	107	57	130	88	77	59	0	610	610	1.00
Obtained	552	611	356	721	509	484	373	594			
Actual	552	611	356	721	509	484	373	594			
Column Factor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00			

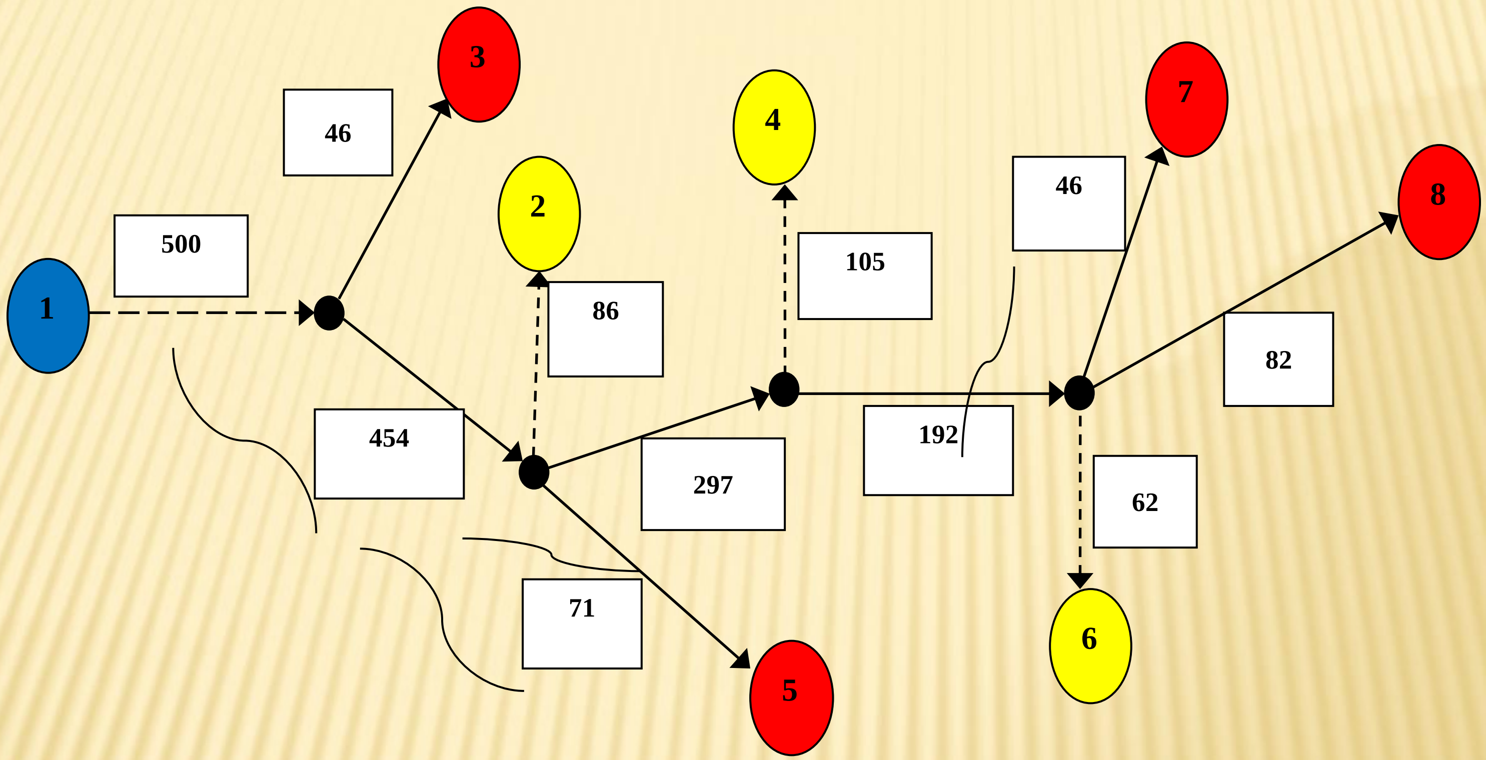

FIGURE 3: TRIP VOLUME FROM ZONE ONE TO OTHER ZONES

TRIP ASSIGNMENT: MINIMUM PATH FINDER (FLOYD'S ALGORITHM)

TABLE 8: SHOWING MATRIX D_0 AND MATRIX S_0 (DISTANCE AND SEQUENCE OF THE NETWORK)

D_0 =

	1	2	3	4	5	6	7	8
1	-	2	4	∞	∞	∞	∞	∞
2	2	-	∞	2	4	∞	∞	∞
3	4	∞	-	5	∞	∞	8	∞
4	∞	2	5	-	4	5	∞	∞
5	∞	4	∞	4	-	7	∞	∞
6	∞	∞	∞	5	7	-	2	5
7	∞	∞	8	∞	∞	2	-	4
8	∞	∞	∞	∞	∞	5	4	-

S_0 =

	1	2	3	4	5	6	7	8
1	-	1	1	1	1	1	1	1
2	2	-	2	2	2	2	2	2
3	3	3	-	3	3	3	3	3
4	4	4	4	-	4	4	4	4
5	5	5	5	5	-	5	5	5
6	6	6	6	6	6	-	6	6
7	7	7	7	7	7	7	-	7
8	8	8	8	8	8	8	8	-

ITERATIONS 1 & 2

Table 9: Iteration k = 1

D_1 =

	1	2	3	4	5	6	7	8
1	-	2	4	∞	∞	∞	∞	∞
2	2	-	6	2	4	∞	∞	∞
3	4	6	-	5	∞	∞	8	∞
4	∞	2	5	-	4	5	∞	∞
5	∞	4	∞	4	-	7	∞	∞
6	∞	∞	∞	5	7	-	2	5
7	∞	∞	8	∞	∞	2	-	4
8	∞	∞	∞	∞	∞	5	4	-

S_1 =

	1	2	3	4	5	6	7	8
1	-	1	1	1	1	1	1	1
2	2	-	1	2	2	2	2	2
3	3	1	-	3	3	3	3	3
4	4	4	4	-	4	4	4	4
5	5	5	5	5	-	5	5	5
6	6	6	6	6	6	-	6	6
7	7	7	7	7	7	7	-	7
8	8	8	8	8	8	8	8	-

Table 10: Iteration k = 2

D_2 =

	1	2	3	4	5	6	7	8
1	-	2	4	4	6	∞	∞	∞
2	2	-	6	2	4	∞	∞	∞
3	4	6	-	5	10	∞	8	∞
4	4	2	5	-	4	5	∞	∞
5	6	4	10	4	-	7	∞	∞
6	∞	∞	∞	5	7	-	2	5
7	∞	∞	8	∞	∞	2	-	4
8	∞	∞	∞	∞	∞	5	4	-

S_2 =

	1	2	3	4	5	6	7	8
1	-	1	1	2	2	1	1	1
2	2	-	1	2	2	2	2	2
3	3	1	-	3	2	3	3	3
4	2	4	4	-	4	4	4	4
5	2	5	2	5	-	5	5	5
6	6	6	6	6	6	-	6	6
7	7	7	7	7	7	7	-	7
8	8	8	8	8	8	8	8	-

ITERATIONS 3 & 4

Table 11: Iteration k = 3

D_3 =

	1	2	3	4	5	6	7	8
1	-	2	4	4	6	∞	12	∞
2	2	-	6	2	4	∞	14	∞
3	4	6	-	5	10	∞	8	∞
4	4	2	5	-	4	5	13	∞
5	6	4	10	4	-	7	18	∞
6	∞	∞	∞	5	7	-	2	5
7	12	14	8	13	18	2	-	4
8	∞	∞	∞	∞	∞	5	4	-

S_3=

	1	2	3	4	5	6	7	8
1	-	1	1	2	2	1	3	1
2	2	-	1	2	2	2	3	2
3	3	1	-	3	2	3	3	3
4	2	4	4	-	4	4	3	4
5	2	5	2	5	-	5	3	5
6	6	6	6	6	6	-	6	6
7	3	3	7	3	3	7	-	7
8	8	8	8	8	8	8	8	-

Table 13: Iteration k = 4

D_4=

	1	2	3	4	5	6	7	8
1	-	2	4	4	6	9	12	∞
2	2	-	6	2	4	7	14	∞
3	4	6	-	5	9	10	8	∞
4	4	2	5	-	4	5	13	∞
5	6	4	9	4	-	7	17	∞
6	9	7	10	5	7	-	2	5
7	12	14	8	13	17	2	-	4
8	∞	∞	∞	∞	∞	5	4	-

S_4=

	1	2	3	4	5	6	7	8
1	-	1	1	2	2	4	3	1
2	2	-	1	2	5	4	3	2
3	3	1	-	3	4	4	3	3
4	2	2	3	-	4	4	3	4
5	2	5	4	5	-	5	4	5
6	4	4	4	6	6	-	6	6
7	3	3	7	3	4	7	-	7
8	8	8	8	8	8	8	8	-

ITERATIONS 5 & 6

Table 14: Iteration k = 5

D_5=

	1	2	3	4	5	6	7	8
1	-	2	4	4	6	9	12	∞
2	2	-	6	2	4	7	10	∞
3	4	6	-	5	9	10	8	∞
4	4	2	5	-	4	5	13	∞
5	6	4	9	4	-	7	17	∞
6	9	7	10	5	7	-	2	5
7	12	10	8	13	17	2	-	4
8	∞	∞	∞	∞	∞	5	4	-

S_5=

	1	2	3	4	5	6	7	8
1	-	1	1	2	2	4	3	1
2	2	-	1	2	2	4	3	2
3	3	1	-	3	4	4	3	3
4	2	4	4	-	4	4	3	4
5	2	5	4	5	-	5	4	5
6	4	4	4	6	6	-	6	6
7	3	3	7	3	4	7	-	7
8	8	8	8	8	8	8	8	-

Table 14: Iteration

D_6 =

	1	2	3	4	5	6	7	8
1	-	2	4	4	6	9	11	14
2	2	-	6	2	4	7	9	12
3	4	6	-	5	9	10	8	15
4	4	2	5	-	4	5	7	10
5	6	4	9	4	-	7	9	12
6	9	7	10	5	7	-	2	5
7	11	9	8	7	9	2	-	4
8	14	12	15	10	12	5	4	-

S_6 =

	1	2	3	4	5	6	7	8
1	-	2	3	2	2	4	6	6
2	1	-	1	2	2	4	6	6
3	3	1	-	3	4	4	3	6
4	2	4	4	-	4	4	6	6
5	2	5	4	5	-	5	6	6
6	4	4	4	6	6	-	6	6
7	6	6	7	6	6	7	-	8
8	6	6	6	6	6	6	8	-

ITERATION 7 & 8

Table 15: Iteration k = 7

D_7=

	1	2	3	4	5	6	7	8
1	-	2	4	4	6	9	11	14
2	2	-	6	2	4	7	9	12
3	4	6	-	5	9	10	8	12
4	4	2	5	-	4	5	7	10
5	6	4	9	4	-	7	9	12
6	9	7	10	5	7	-	2	5
7	11	9	8	7	9	2	-	4
8	14	12	12	10	12	5	4	-

S_7=

	1	2	3	4	5	6	7	8
1	-	2	3	2	2	4	6	6
2	1	-	1	2	2	4	6	6
3	3	1	-	3	4	4	3	7
4	2	4	4	-	4	4	6	6
5	2	5	4	5	-	5	6	6
6	4	4	4	6	6	-	6	6
7	6	6	7	6	6	7	-	7
8	6	6	7	6	6	8	8	-

Table 16: Iteration k = 8

D_8=

	1	2	3	4	5	6	7	8
1	-	2	4	4	6	9	11	14
2	2	-	6	2	4	7	9	12
3	4	6	-	5	9	10	8	12
4	4	2	5	-	4	5	7	10
5	6	4	9	4	-	7	9	12
6	9	7	10	5	7	-	2	5
7	11	9	8	7	9	2	-	4
8	14	12	12	10	12	5	4	-

S_8=

	1	2	3	4	5	6	7	8
1	-	1	1	2	2	4	6	6
2	2	-	1	4	5	4	6	6
3	3	1	-	3	4	4	3	7
4	2	4	4	-	4	4	6	6
5	2	5	4	5	-	5	6	6
6	4	4	4	6	6	-	6	6
7	6	6	3	6	6	7	-	7
8	6	6	7	6	6	8	8	-

TABLE 17: RECOMMENDED SHORTEST PATH AND TRIP VOLUMES

NODE					
FROM	TO	MINIMUM LINK PATH	TRAVE TIME (Min)	TRIPS	LINK VOLUME
1	2	1 - 2	2	86	454
	3	1 - 3	4	46	46
	4	1 - 2 - 4	4	105	297
	5	1 - 2 - 5	6	71	71
	6	1 - 2 - 4 - 6	9	62	192
	7	1 - 2 - 4 - 6 - 7	11	48	48
	8	1 - 2 - 4 - 6 - 8	14	82	82
2	1	2 - 1	8	100	162
	3	2 - 1- 3	6	62	62
	4	2 - 4	2	141	402
	5	2 - 5	4	96	96
	6	2 - 4 - 6	7	84	261
	7	2 - 4 - 6 - 7	9	65	65
	8	2 - 4 - 6 - 8	12	112	112
3	1	3 - 1	4	49	106
	2	3 - 1- 2	6	57	57
	4	3 - 4	5	69	157
	5	3 - 4 - 5	9	47	47
	6	3 - 4 - 6	10	41	41
	7	3 - 7	8	32	87
	8	3 - 7 - 8	12	55	55
4	1	4 - 2 - 1	4	115	115
	2	4 - 2	2	134	249
	3	4 - 3	5	71	71
	5	4 - 5	4	110	110
	6	4 - 6	5	97	300
	7	4 - 6 - 7	7	75	75
	8	4 - 6 - 8	10	128	128

TABLE 17 CONT…

5	1	5 - 2 - 1	6	95	95
	2	5 - 2	4	109	204
	3	5 - 4 - 3	9	58	58
	4	5 - 4	4	133	191
	6	5 - 6	7	79	245
	7	5 - 6 - 7	9	61	71
	8	5 - 6 - 8	12	105	105
6	1	6 - 4 -2 - 1	9	49	49
	2	6 - 4 - 2	7	57	106
	3	6 - 4 -3	10	30	30
	4	6 - 4	5	69	205
	5	6 - 5	7	47	47
	7	6 - 7	2	33	33
	8	6 - 8	5	55	55
7	1	7- 6 - 4 - 2 -1	11	52	52
	2	7- 6 - 4 - 2	9	61	113
	3	7 - 3	8	32	32
	4	7 - 6 - 4	7	74	187
	5	7 - 6 - 5	9	50	50
	6	7 - 6	2	44	281
	8	7 - 8	4	57	57
8	1	8 - 6- 4 - 2 -1	14	92	92
	2	8 - 6 - 4 - 2	12	107	199
	3	8 - 7 - 3	12	57	57
	4	8 - 6 - 4	10	130	329
	5	8 - 6 - 5	12	88	88
	6	8 - 6	5	77	494
	7	8 - 7	4	59	116

CONCLUSION

This study can be generalized to other means of transportation apart from motorcycles, and the methodology can be adopted by other places with modification in respective data. The study provides both present and potential motorcyclists and other road users with minimum link paths and link flow volumes in Bida town. Table 17 furnishes us with the information as a recommendation which can practically be implemented. The optimized method retains the characteristic of simple calculation in the all-or-nothing algorithm. Owing to the consideration of road network conditions in the assignment process, the assignment results are more practical and can adapt to the characteristics of the town system, such as complicated road network and heavy traffic volume.

REFERENCES

Calum M. C. (2012). Dispensing Charity: The Deficiencies of an All-or-Nothing Fiscal Concept. *VOLUNTAS: International Journal of Voluntary and Nonprofit Organizations*, 23(2), 392-414

Chen B. (2012). All-or-nothing payments. *Journal of Mathematical Economics*, 48(3), 133-142

Hui C. (2014). *Application Study of All-or-Nothing Assignment Method for Determination of Logistic Transport Route in Urban Planning.* Computer Modelling & New Technologies. 18(12C); 932-937

Priyanto S. and Friandi E. P. (2010). Analyzing of Public Transport Trip Generation in Developing Countries (A Case Study in Yogyakarta, Indonesia)

Taha H. (2010). Introduction to Operations Research. Prentice Hall. USA

Tom V.M. (2006). Transportation Network Design, http://www.civil.iitb.ac.in

Wardrop, J.G. (1952). Some Theoretical Aspects of Road Traffic Research.*Proceedings ofthe Institute of Civil Engineers*, Part II, vol. 1, pp. 325-378.

Yan-Bing L., Qiao-Yan W., Su-Juan Qin, F. and Ying S. (2014). Practical quantum all-or-nothing oblivious transfer protocol. *Quantum Information Processing*, 13(1), 131-139

THANK YOU VERY MUCH FOR YOUR TIME