OPTIMIZING CITY BUS ROUTES: MINIMIZING TOTAL TRAVEL TIME WHILE MAXIMIZING NUMBER OF STOPS, A CASE STUDY OF THE CITY OF CHILLICOTHE.

1.0 INTRODUCTION

Bus Routing Problem is one of the most widely studied topics in the field of Operations Research. The bus routing problem is a classic problem in transportation planning that involves determining the optimal routes and schedules for a fleet of buses to efficiently serve a given set of bus stops while meeting various constraints and objectives. The problem is particularly important in urban areas, where buses are a critical mode of public transportation for many residents and workers. Effective bus routing can improve accessibility and reduce traffic congestion, as well as promote sustainable and equitable urban development.

The bus routing problem is complex due to the large number of variables involved, including the number and locations of bus stops, the frequency and timing of bus services, the capacity of buses, and the demand for transportation. The problem also involves trade-offs between competing objectives, such as minimizing travel time for passengers, maximizing ridership, minimizing operating costs, and minimizing environmental impacts. Various techniques have been developed to address the bus routing problem, including mathematical optimization models, heuristic algorithms, and simulation-based approaches. These techniques can be applied to a range of specific scenarios, from simple fixed-route systems to more complex demand-responsive or flexible-route systems.

2.0 LITERATURE REVIEW

Most of the literature on the vehicle routing problem has been ordinary vehicle routing, truck routing for distribution, and school bus routing, only a limited amount of research has considered local or community bus routing. In the literature on routing of local or community buses, only a small number of papers considers the simultaneous selection of stops and minimizing total travel time. Most community bus vehicle routing formulations focus on formulating schedule for a fleet of busses, minimizing total number of busses needed, maximizing capacity of busses, and several of its extensions (e.g., time windows).

In Shafahi & Haghani (2017), they proposed a new formulation for the multi-school homogeneous fleet routing problem that maximizes trip compatibility while minimizing total travel time. The plan incorporated a trip compatibility for scheduling in the routing problem and proposed heuristic algorithm for its solution. To compare the performance of the model with traditional routing problems, eight midsize data sets were generated, and the results showed that the proposed model can reduce the buses needed by up to 25%.

Also, Li & Fu (2002), minimized the total bus travel time using a test data from a kindergarten in Hong Kong. The paper proposed a heuristic algorithm for its solution which measures the total travel time of the bus. The numerical result showed to be effective with a saving of 29% in total travelling times when comparing to current practice.

Vaughan & Cousins (1977) Studied the travel time on a bus route minimizing the average travel time with respect to the distribution of stops along the route, given a budget which determines the total number of stops. The paper used a continuous function to describe the joint distribution of origins and destinations. It concluded that the travel time route where travel is possible in both directions is calculated as a function of density of stops at any given point.

Kuo et al. (2013) designed a bus route to satisfy the demands of most passengers within a limited total bus travel time. They optimized parameters setting using the Taguchi method and proposed a Simulated Annealing (SA) algorithm optimizing the routing design. The experimental results show that the proposed SA algorithm with the optimal parameters setting results in better routes than those designed by other research methods.

3.0 METHODOLOGY

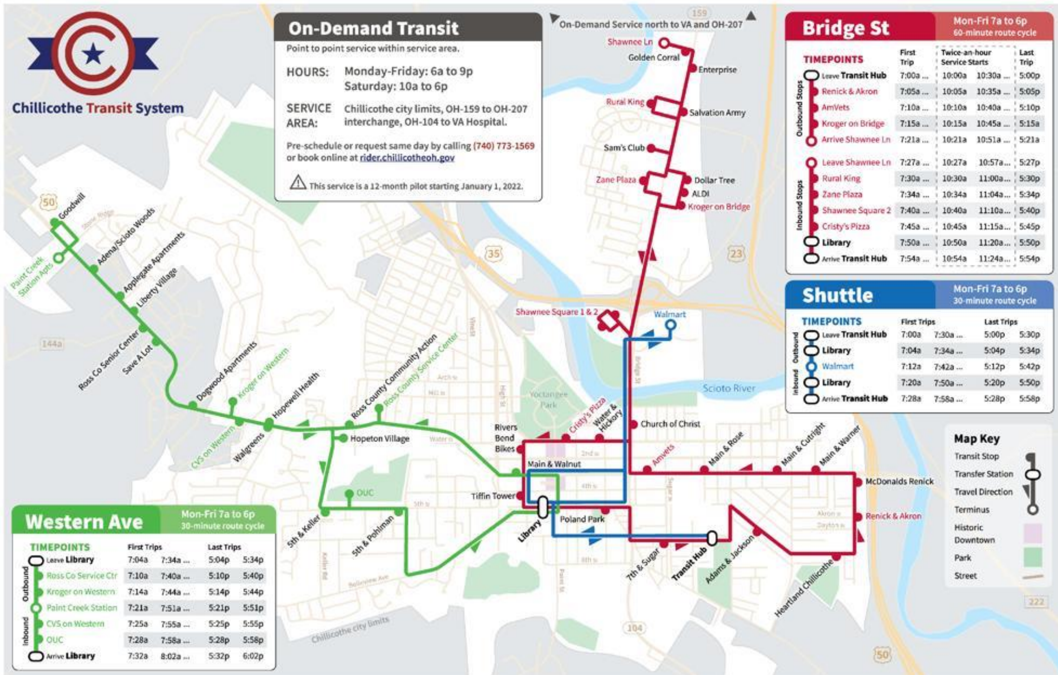

Figure_1. *Western Avenue – The green colored route*

3.1 Problem statement.

"The issue we struggle with is doing the Western Avenue fixed route loop in a half an hour, which is what the route is supposed to be. According to the scheduled timetable, the bus should leave the library (starting point) at four minutes on top each hour and half hour, visit each of the stop points on its route once and return the library. If there are no delays, this can be done with little to no problem. However, if there is traffic, a person or persons who needs to utilize the lift, any vehicles blocking the turnaround area at Hopeton Village, or even worse a combination of any of these, it can put the driver behind schedule. Once this happens, it is hard to get back on the expected schedule, as the time frame is so tight. We wish to minimize the total travel time of the route loop to at most half an hour as the route is expected to be."

3.2. Mathematical Model

Many researchers have developed mathematical optimization models to solve the bus routing problem. These models typically involve defining an objective function that balances competing goals such as minimizing passenger travel time, minimizing operating costs, and maximizing bus utilization. The models also incorporate constraints such as vehicle capacity, time windows for arrival and departure at bus stops, and traffic congestion. Researchers have used a variety of mathematical programming techniques, such as integer programming, mixed-integer programming, and dynamic programming, to solve these models.

3.2.1. Problem representation.

Consider the set ‘STOPS’ to be the set of all the stop points on the western avenue route, there are 21 stops on the route. The input parameters are the following.

‘ttime’ is the time it takes to travel from stop point i to stop point j, google map was used to calculate these travel times between stops.

‘wtime’ is the time spent at each stop point i due to loading and unloading passengers, it is estimated based on existing data.

‘required’ is a binary parameter which is 1 if a stop point is required and equal 0 if it is optional. Some of the stops on the route are at a walking distance from one another. Some of these adjacently closed stops are made optional, so that we can ignore to meet the time goal.

‘link’ is also a binary parameter which is equal 1 if it is possible to travel from stop point i to stop point j and equal 0 if otherwise. This is to avoid paths which are not possible due to traffic rules, for instance one ways.

‘incl’ is a binary decision variable which is expected to equal 1 if a stop is included on the route and equal 0 if otherwise. Since some stops are optional the model will decide which stop to include on the route to meet the time goal.

'path' is another binary decision variable expected to equal 1 if stop i is reached from stop j (i.e., if arc i-j is included), and equal 0 if otherwise, given that it is possible to travel from stop point i to stop point j (the **'link'** parameter equal 1).

3.2.2. Objective function

1. The objective is this project is to reduce the bus's total travel time spent to make one complete trip (tour) on the western avenue route to a half an hour. As stated above in the problem statement, the City of Chillicothe wants the western avenue bus route to be done in 30 minutes to speed up movement of passengers traveling by the public transport along the route.

```
minimize totaltime: sum {i in stops} wtime[i]*incl[i] + sum{ i in stops, j in stops:
link[i,j] = 1} ttime [i,j]* path[i,j]
```

2. Because there are so many stops to visit on the western avenue route, the second objective is to maximize the number of stop points visited on the western avenue route as many as possible while still doing the route within half an hour always.

```
maximize number_of_stops: sum {i in stops} incl[i];
```

3.2.3. Constraints

Three major constraints are considered in this project in attempt to achieve our objective.

1. Total time constraint: as stated in the objective, the total travel time spent to complete one trip is limited to half an hour.
2. Conservation of flow: There must be only one arc entering a stop point and only arc also leaving the same stop point. This constraint is to make the solution result in a tour. That is, the bus start from one stop points, visit as many stops as possible and return to the starting point.

3. The bus must stop at each required stop point once and stop at the optional stop point either once or not stopping at all in a single trip.

3.3. Solution Method

The problem can be transformed into a Travelling Salesman Problem (TSP) and solved by using one of the available algorithms. In multi-objective problems, it is very difficult to find a solution which optimizes all objectives. In the case of this problem, one of the objectives was added to the constraints. This makes it easy to find an optimal solution. The problem was solve using AMPL software. Branch and bound solution algorithm were used by AMPL Cplex solver to obtain an optimal solution for the problem. From our class website, we learnt that the branch-and-bound technique is to *divide and conquer*. Since the original “large” problem is hard to solve directly, it is *divided* into smaller and smaller subproblems until these subproblems can be *conquered*. The *dividing (branching)* is done by partitioning the entire set of feasible solutions into smaller and smaller subsets. The *conquering (fathoming)* is done partially by giving a *bound* for the best solution in the subset and discarding the subset if the bound indicates that it cannot contain an optimal solution.

4.0 RESULTS AND ANALYSIS

4.1. Data input

Google map was used to collect data on the travel time between the stops on the western avenue route. Wait times were derived from the existed data on the route. It was estimated by averaging the various times that different drivers had spent at each of the stops. Selection of required and optional stops were made based on proximity and the volume of pickup and drop off at each stop. Stops with less volume of pickup and drop off and are very closed to other major stops are made optional, while these major stops with high volume of pickup and drop off are required. The link parameter was determined using traffic rules and location of stops. For example, stops of the opposite sides of the road cannot connect.

4.2. Analysis

The initial data collected resulted in an infeasible solution. The initial travel times data collected were high, because as at the time of collecting the data, there was a traffic jam which to a go-slow on the western avenue. This made Google map provided high travel times even between the closest stops. The route could not be completed in within 30 minutes. The time was increased to 50 minutes, but it was still infeasible. The travel time had to be recollected at an appropriate time with the day.

The solution obtained after the travel time data were recollected and wait times were adjusted resulted in subtours; Different groups of stops were connected to form individual tours isolated from the other groups. This is shown by the 'red ink' on *figure_2* below. For instance, ***lib - main*** formed one subtour, ***adna – gdwl - ptsca***, formed a subtours, ***cvs - wlgrn - ouc - fkler*** also formed another subtour, among others. There were about six different subtours, meanwhile the solution is expected to have a single tour that visit as many stops as possible and returns to the origin.

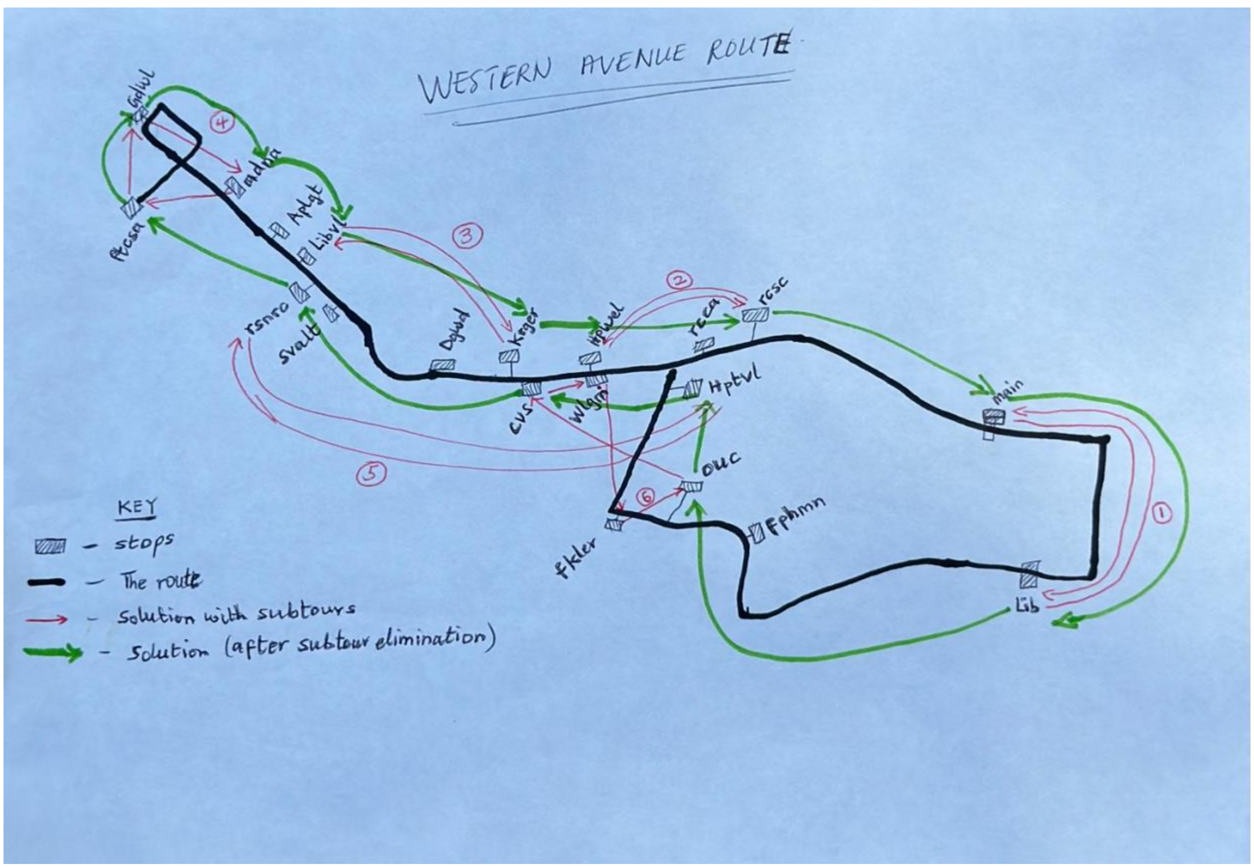

Figure_2

The problem of subtours was resolved using subtour elimination constraints. These constraints requires that each subtours must have at least one arc joining the other subtours, so that no single group of stops is isolated. This made it possible to connect all the subtours to form a single tour, which is indicated by the 'green ink' as shown in figure_2 above. The green ink route is figure_2 above is the current proposed solution to the Western Avenue bus routing problem. It can be completed within 30 minutes as expected. The bus starts from the library stop point (**lib**) visit 14 stop points in a clockwise direction and return the library.

5.0 FUTURE WORKS

The project can be continued by looking at the bus capacity, demand for transportation as additional constraints. The objectives can also be modified, such as minimizing travel time for passengers, maximizing ridership, minimizing operating costs, and minimizing environmental impacts. The ongoing researchs in this area continues to push the boundaries of transportation planning and to provide new insights and solutions to the challenges of urban mobility.

6.0 CONCLUSION

In conclusion, the minimization of travel time bus routing problem is a critical transportation planning challenge that has been the subject of extensive research over the years. Efficient and reliable bus routing can significantly improve the accessibility and mobility of residents and workers in urban areas, while also reducing traffic congestion and promoting sustainable development. In this paper, a case study of the City of Chillicothe western avenue bus routing problem is described. It is formulated as a multi-objective combinatorial optimization problem. A branch and bound algorithm were used by the AMPL software for its solution as shown in *figure_2* with 'green ink'. The model developed, runs efficiently on a PC. The objectives of the western avenue bus routing problem considered include minimizing the total travel time to half an hour and maximizing the number of stops visited which the city was expecting. The ongoing research in this area continues to push the boundaries of transportation planning and to provide new insights and solutions to the challenges of urban mobility.

7.0 REFERENCE

Shafahi, A., Wang, Z., & Haghani, A. (2017). Solving the school bus routing problem by maximizing trip compatibility. *Transportation Research Record*, *2667*(1), 17-27.

Li, L. Y., & Fu, Z. (2002). The school bus routing problem: a case study. *Journal of the Operational Research Society*, *53*, 552-558.

Vaughan, R. J., & Cousins, E. A. (1977). Optimum location of stops on a bus route. In *International Symposium on Transportation and Traffic Theory, 7th, 1977, Kyoto, Japan*.

Kuo, Y., Luo, C. M., & Wang, C. C. (2013). Bus route design with limited travel time. *Transport*, *28*(4), 368-373.

Class website; https://people.ohio.edu/melkonia/math4630/slides.html

APPENDIX

Model;

```
set stops;

set s1;
set s2;
set s3;
set s4;
set s5;
#set s6;
#set s7;


param ttime {stops, stops}; # travelling time from stop i to j

param wtime {stops}; # includes loading, unloading, and turnaround time

param required {stops} binary; # 1 if a stop is not required and 0 otherwise

param link {stops, stops} binary; # 1 if it's possibe to move from stop i to j and 0
otherwise
```

```
var incl {i in stops} binary; # 1 if stop i is included in the route

var path {i in stops, j in stops: link[i,j] = 1} binary; #1 if stop j is reached from stop i on route, and 0 otherwise.


maximize number_of_stops: sum {i in stops} incl[i];

s.t. totaltime: sum {i in stops} wtime[i]*incl[i] + sum{ i in stops, j in stops: link[i,j] = 1} ttime [i,j]* path[i,j] <= 1800 ;

s.t. conservation_of_flow {i in stops}: sum{j in stops: j != i and link[i,j] = 1} path[i,j] = sum{j in stops: j != i and link[j,i] = 1} path[j,i];

s.t. enter_each_required_stop{i in stops: required[i] = 1}: sum{j in stops: j != i and link[j,i] = 1} path[j,i] = 1;

s.t. might_enter_each_optional_stop{i in stops: required[i] = 0}: sum{j in stops: j != i and link[j,i] = 1} path[j,i] <= 1;

s.t. var_connection {i in stops, j in stops: link[i,j] = 1} : path[i,j] <= incl[i];

s.t. subtour1: sum{i in s1, j in stops diff s1: link[i,j] = 1} path[i,j] >= 1;

s.t. subtour2: sum{i in s2, j in stops diff s2: link[i,j] = 1} path[i,j] >= 1;

s.t. subtour3: sum{i in s3, j in stops diff s3: link[i,j] = 1} path[i,j] >= 1;
```

```
s.t. subtour4: sum{i in s4, j in stops diff s4: link[i,j] = 1} path[i,j] >= 1;

s.t. subtour5: sum{i in s5, j in stops diff s5: link[i,j] = 1} path[i,j] >= 1;



DATA;

data;
set stops := lib main rcsc rcca hpwel krger dgwd libvl aplgt adna gdwl ptcsa rsnrc svalt
cvs wlgrn hptvl fkler ouc fphmn ;
set s1 := lib main;
set s2 := rcsc rcca;
set s3 := hpwel krger libvl;
set s4 := aplgt adna gdwl ptcsa rsnrc;
set s5 := cvs hptvl fkler ouc;
#set s6 := rsnrc cvs hptvl;
#set s7 := fkler ouc;

param ttime :
                    lib         main  rcsc  rcca  hpwel krger dgwd  libvl  aplgt
     adna  gdwl     ptcsa rsnrc svalt cvs         wlgrn hptvl fkler ouc
     fphmn :=
lib                 0           60          240         240         240
     250            300         350         420         400         410
     490            310         320         280         290         285
     205            205         190
```

main	60	0	160	165	170	205
	280	295	310	330	350	420
	280	290	200	160	240	170
	200	100				
rcsc	205	160	0	60	100	160
	190	205	210	280	280	310
	200	205	160	120	240	120
	160	165				
rcca	200	160	60	0	60	160
	160	165	200	210	220	315
	160	160	100	60	160	105
	100	160				
hpwel	280	170	100	105	0	55
	60	100	110	160	170	280
	100	105	55	60	160	100
	100	160				
krger	310	200	160	160	55	0
	100	105	160	160	170	285
	100	105	205	60	200	160
	160	200				
dgwd	360	310	200	200	100	100
	0	100	105	160	165	210
	100	55	100	105	280	200
	200	280				
libvl	360	310	205	200	100	10
	105	0	55	60	60	100

55 55 100 100 280 200
280 275
aplgt 360 310 200 200 160 160
100 55 0 60 55 100
60 45 100 160 280 200
275 270
adna 400 310 250 230 180 185
145 55 45 0 45 80
55 45 140 180 285 250
310 310
gdwl 780 720 660 660 480 480
480 360 360 360 0 350
350 370 470 510 720 660
720 720
ptcsa 600 540 420 360 360 300
240 160 100 100 100 0
160 160 280 280 410 360
720 480
rsnrc 420 360 240 240 120 120
120 60 60 60 60 160
0 55 100 100 100 200
280 280
svalt 420 300 240 240 120 120
120 60 60 60 60 160
55 0 100 100 280 200
280 280

cvs 310 280 160 160 55
60 120 160 160 160 200
280 100 100 0 60 205
160 160 200

wlgrn 280 200 100 100 55 60
100 160 160 200 200 280
160 100 55 0 160 100
160 160

hptvl 205 155 55 60 60 100
100 160 160 200 205 280
160 160 100 60 0 160
200 205

fkler 280 165 100 100 100 100
160 200 205 280 280 310
160 160 100 100 60 0
60 60

ouc 280 205 100 100 100
160 160 200 200 280 280
310 200 205 100 100 100
60 0 100

fphmn 160 100 160 160 160
205 200 310 280 310 310
360 310 200 160 160 100
60 60 0

```
param wtime :=
lib         40
main        5
rcsc        10
rcca        5
hpwel       40
krger       15
dgwd        5
libvl       10
aplgt       5
adna        5
gdwl        15
ptcsa       5
rsnrc       0
svalt       5
cvs         5
wlgrn       5
hptvl       40
fkler       5
ouc         10
fphmn       5

;

param required :=
lib         1
main        1
```

rcsc	1
rcca	0
hpwel	1
krger	1
dgwd	0
libvl	1
aplgt	0
adna	1
gdwl	1
ptcsa	1
rsnrc	1
svalt	0
cvs	1
wlgrn	0
hptvl	1
fkler	0
ouc	1
fphmn	0

;

param link :

lib main rcsc rcca hpwel krger dgwd libvl aplgt

adna gdwl ptcsa rsnrc svalt cvs wlgrn hptvl fkler ouc

fphmn :=

lib 0 1 1 1 1
1 1 0 0 0 0
0 0 0 1 1 1
1 1 1

main 1 0 1 1 1 1
1 1 1 1 1 0
0 0 0 0 1 1
1 1

rcsc 1 1 0 1 1 1
1 1 1 1 1 0
0 0 0 0 0 0
0 0

rcca 1 1 1 0 1 1
1 1 1 1 1 0
0 0 0 0 0 0
0 0

hpwel 1 1 1 1 0 1
1 1 1 1 1 1
0 0 0 0 0 0
0 0

krger 1 1 1 1 1 0
1 1 1 1 1 1
0 0 0 0 0 0
0 0

dgwd 1 1 1 1 1 1
0 1 1 1 1 1

0 0 0 0 0 0
0 0

libvl 0 1 1 1 1 1
1 0 1 1 1 0
0 0 0 0 0 0
0 0

aplgt 0 1 1 1 1 1
1 1 0 1 1 1
0 0 0 0 0 0
0 0

adna 0 1 1 1 1 1
1 1 1 0 1 1
0 0 0 0 0 0
0 0

gdwl 0 0 0 0 1 1
1 1 1 1 0 1
1 1 1 0 0 0
0 0
ptcsa 0 0 0 0 0 0
0 0 0 0 1 0
1 1 1 1 1 0
0 0

rsnrc 0 0 0 0 0 0 0

0 0 0 1 1 0

1 1 1 0 1 1

1

svalt 0 0 0 0 0 0

0 0 0 0 1 1

1 0 1 1 1 1

1 1

cvs 1 0 0 0 0

0 0 0 0 0 1

1 1 1 0 1 1

1 1 1

wlgrn 1 0 0 0 0 0

0 0 0 0 0 1

1 1 1 0 1 1

1 1

hptvl 1 0 0 0 0 0

0 0 0 0 0 1

0 1 1 1 0 1

1 1

fkler 1 0 0 0 0 0

0 0 0 0 0 1

1 1 1 1 1 0

1 1

ouc 1 0 0 0 0

0 0 0 0 0 0

1	1	1	1	1	1
1	0	1			
fphmn	1	0	0	0	0
0	0	0	0	0	0
0	1	1	1	1	1
1	1	0			