

INGENIERÍA DEL CONOCIMIENTO (2020-2021)
MÁSTER EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Lógica difusa aplicada a los semáforos

Manuel Orantes Taboada, Carlos Sanchez Martinez

1 de febrero de 2021

Índice

1	Resumen	3
2	Introducción	3
3	Diferentes estudios sobre los controladores de semáforos con lógica difusa	3
4	Otros trabajos relacionados	6
5	Diseño de un simulador de tráfico para un cruce	7
6	Modelo	8
7	Modelo de lógica difusa	11
8	Uso de los semáforos inteligentes alrededor del mundo	20
8.1	Pittsburg, Pennsylvania (USA)	20
8.2	Artículo en MESinsights	21
8.3	NoTraffic	21

1. Resumen

Actualmente los semáforos de todo el mundo funcionan con ciclos fijos de tiempo pre-determinado. Desde que se definieran los principios de lógica difusa, muchos han sido los trabajos que han enfocado este tema, y se han realizado aproximaciones con lógica difusa por todo el mundo. Uno de los principales trabajos fue el propuesto por Pappis y Mamdani, y muchos otros siguieron sus pasos a lo largo de los últimos años. La idea del funcionamiento del controlador es la siguiente: a partir de unas reglas que se han sacado a base de ensayo y error, se calcula el grado de confianza para clasificar el tiempo de retraso de los coches, siendo la llegada de estos aleatoria. Con todos estos datos se consigue minimizar el tiempo de espera de los vehículos. Hoy en día se siguen desarrollando tanto teorías como sistemas que en un futuro nos permitan vivir sin tantos atascos, consiguiendo a su vez reducir el tiempo de espera de los conductores, y la emisiones producidas por los mismos.

Keywords: lógica difusa, cruce de carreteras , semáforo inteligente, controlador difuso

2. Introducción

La mayoría de las señales de tráfico de hoy en día están controladas por ciclos fijos de tiempo, es decir, el semáforo está en verde durante x segundos y después está en rojo durante y segundos, siendo normalmente distintos x e y . Esto es una solución muy simple, pero que no suele ayudar en la congestión del tráfico, haciendo que esta no sea ni mucho menos la óptima. Pero ha habido muchos avances en la tecnología, y los semáforos inteligentes no deben de ser ignorados, sobre todo en las grandes urbes que mueven inmensas cantidades de tráfico.

3. Diferentes estudios sobre los controladores de semáforos con lógica difusa

Desde que Zadeh(1921-2017) definiera en 1965 los conceptos de la lógica difusa, esta se ha usado en multitud de proyectos, sobre todo los relacionados con la ingeniería. Entre ellos, se han realizado muchos estudios de la aplicación de la lógica difusa a los semáforos.

El más popular fue el que propusieron Pappis y Mamdani [2]. Construyeron un modelo para un cruce de dos vías en donde cada vía solo tenía un carril. Después, fueron sucesivos trabajos los que se fueron presentando, entre ellos destacaremos los de Nakatsuyama [3] o Favilla [4], que ya introducían múltiples carriles en el modelo. Incluso en un trabajo de Trabia [5] se simuló un cruce con cuatro carriles, incluyendo la opción de que los coches giraran a la derecha en el cruce. Hay que especificar que, en este último paper, al igual que en otros como el de Pappis y Mamdani [2], se hizo el estudio en Reino Unido, y lo que se permitía es que los coches giraran a la izquierda, debido a que conducen en sentido opuesto a lo que estamos acostumbrados. Para facilitar la comprensión, se han unificado

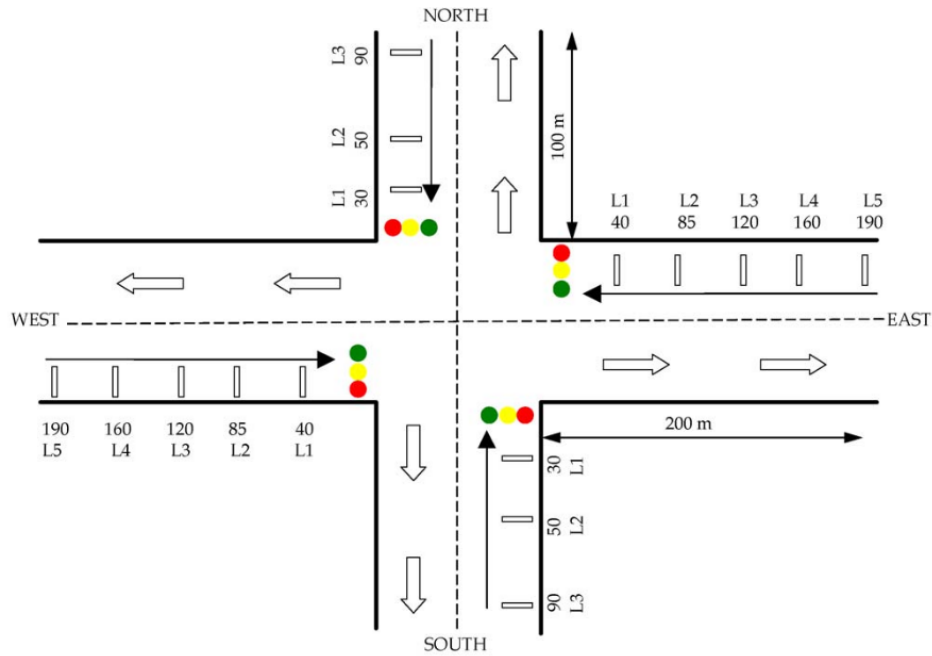
todos estos criterios tomando de referencia que se circula por la derecha, y por lo tanto, cambiamos el sentido de los papers realizados en Inglaterra.

Después, hubo otra vertiente de trabajos cuyo objetivo era minimizar el tiempo de espera, hasta obtener el óptimo. Ella Bingham [6] usó la misma configuración que usaron Pappis y Mamdani [2]. En su estudio, los parámetros que definía el controlador del semáforo de Mamdani fueron determinados mediante aprendizaje por refuerzo basado en simulaciones. En su trabajo, ella misma comentó que conseguía un gran resultado con volúmenes de tráfico constantes. Chou y Teng [7] extendieron el controlador de las señales de tráfico para resolver el problema de varios cruces consecutivos. Su controlador podía ser usado hasta con colas de más de 700 metros. Disponían de diferentes sensores a lo largo del recorrido para obtener la información del tráfico. Usaban una ecuación parabólica, debido a su simplicidad, para simular el tránsito de los vehículos. Su objetivo era construir un simulador de tráfico que se asemejara al del mundo real. Usaban la longitud máxima de la cola de coches para las direcciones relativas de las intersecciones. Notar que en su trabajo usaron longitudes diferentes a las que se usaron en los primeros papers de los que hemos hablado.

Chong [8] investigó los resultados de 4 redes neuronales difusas como arquitecturas distintas para un controlador de semáforo para un simple cruce de 4 direcciones. El entrenamiento de las redes neuronales difusas fue realizado con datos recogidos de un simulador de tráfico por un operario que sí era humano. Indicó que todas las arquitecturas probadas realizaban un buen trabajo. Schmöcker [9] en cambio realizó un estudio en el que prestó un especial interés en la espera de los peatones en los cruces de cuatro direcciones.

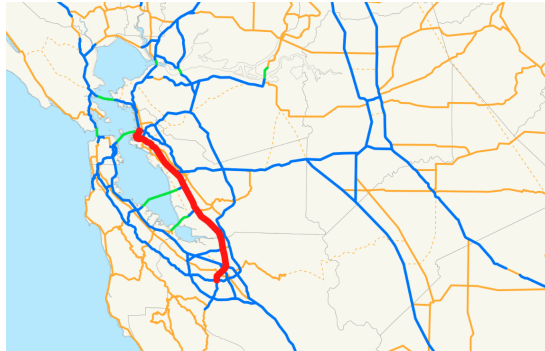
Como podemos observar, son muchos los estudios que se han realizado en este campo, con multitud de configuraciones. Los cambios más comunes siempre han sido tener en cuenta uno o más cruces o la cantidad de carriles de estos. También, como hemos visto, se han tenido en cuenta a los peatones o si en el cruce se podía girar a la derecha o no. Además, se han probado multitud de técnicas diferentes, aunque todas tienen como base a la lógica difusa.

Lo que sí tienen todos en común, es que para realizar el estudio se ha creado un simulador de tráfico para poder tener una referencia de cómo se están comportando los controladores de los semáforos en un ambiente preparado para asemejarse lo más posible a la realidad. Veamos ahora un ejemplo de como podría funcionar un simulador de tráfico para un cruce como el siguiente:



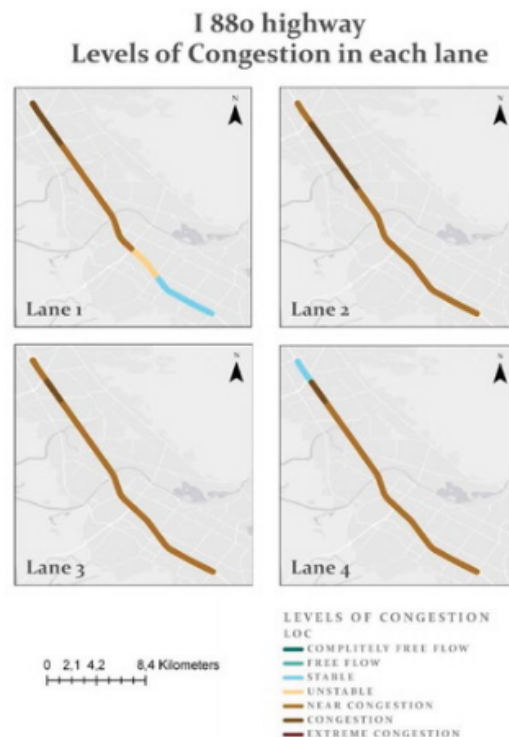
4. Otros trabajos relacionados

Como acabamos de ver, en los trabajos se estudia como mejorar la eficacia de los ciclos de los semáforos a partir de un flujo de tráfico, que casi siempre es proporcionado por distintos simuladores de tráfico. Pero hay algunos estudios que también usan lógica difusa que se encargan de eso mismo, de calcular si hay congestión de tráfico o no. Este es el caso de un reciente estudio de Kalinic [10] (2019). En el se busca justo eso, ver por medio de técnicas difusas el estado de una de las carreteras de Estados Unidos, más concretamente de la interestatal 880, de California.



En el trabajo se dividía la carretera en 6 trozos, y se usaban los siguientes conjuntos difusos:

- Completamente libre de circulación.
- Libre de circulación.
- Circulación estable.
- Circulación inestable.
- Circulación cerca de congestionarse.
- Circulación congestionada.
- Circulación extremadamente congestionada.



En las conclusiones a las que llegaron, destacamos que se diferenciaban de los métodos clásicos de cálculo de congestión de carreteras, que ellos no usaban el número de coches en relación a un determinado segmento de la carretera. En cambio, su aproximación estaba basada en el lenguaje natural y el sentimiento de los conductores implicados en cada sector. Además, declaran que su método es sencillo de aplicar y sigue la lógica del sentido común, y que en un futuro le añadirían nuevas reglas referentes a la velocidad de los vehículos.

5. Diseño de un simulador de tráfico para un cruce

Vamos a ilustrar que sería necesario para la creación de un simulador de tráfico, usando unos datos ya dados según la anterior imagen. Para ello, los principales puntos que se deben de tener en cuenta son los siguientes:

- La velocidad media que se da de una dirección a otra cuando el semáforo está en verde. En nuestro caso, tomaremos la velocidad media de los coches que van de oeste a este como 12 m/s.
- El tiempo necesario para alcanzar el semáforo desde los 200 metros en donde están situados los detectores. Asumiendo que no habrá ningún coche en cola, podemos hallar este tiempo usando la siguiente fórmula:

$$t = \frac{\Delta x}{v} = \frac{200m}{12m/s} = 16,66s$$

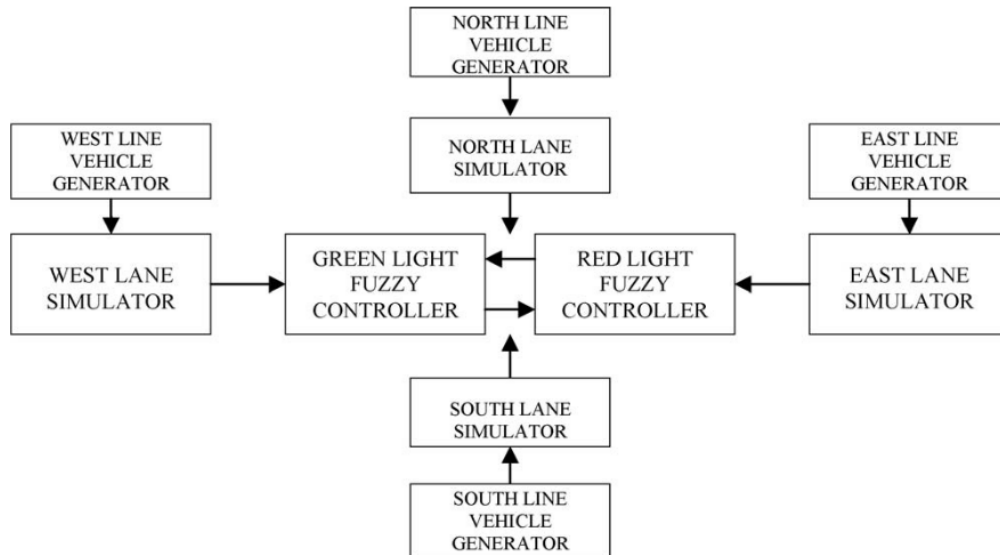
- De manera similar situamos en 12m/s la velocidad media de un vehículo que circula de norte a sur y sabemos que tardará 8.33 segundos en recorrer los 100 metros que separan al semáforo de los sensores.

Es muy importante remarcar que la distancia de los sensores a los semáforos serán muy importantes a la hora de determinar el mínimo de la duración de un semáforo con estado abierto, es decir, con la luz verde. La duración del semáforo abierto o cerrado (es decir, en verde o en rojo) debe de modificarse teniendo en cuenta la cantidad de coches en cola en cada momento, para intentar minimizar el tiempo que los coches esperar para cruzar. Por ejemplo, si tenemos una duración de un semáforo abierto de entre 20 y 120 segundos, cuando no haya ningún coche para pasar y en la otra dirección varios coches estén guardando cola, el semáforo deberá cerrarse para dejar pasar a los coches de la otra dirección. Además, debe de haber un valor máximo también sobre el cuál no pueda exceder el tiempo que un semáforo está abierto.

En los semáforos convencionales, la longitud de los vehículos no suele ser tomada en cuenta a la hora de ver la duración de las fases de los semáforos. A pesar de esto, la longitud real de los vehículos (que oscila desde los 4 metros hasta los 8) si es tomada en algunos estudios, como es el caso del paper de Chou y Teng [7]. En otros estudios, los sensores han sido también utilizados para calcular el número de vehículos en cada carril y saber

también la longitud de los vehículos aplicados, para dar una respuesta acorde de los estados de los semáforos.

Con todo esto, presentamos en la siguiente imagen como interactuarán cada una de las partes implicadas. Existen 4 generadores de vehículos, uno en cada dirección, que confluyen en el cruce, el cual está controlado por el simulador del semáforo.



En la realidad, los datos reales pueden adquirirse poniendo diferentes detectores en los carriles próximos a la intersección, en intervalos propicios. Si estos datos son imposibles de obtener por la fisionomía de la intersección, o por el presupuesto escaso, debemos de poder obtener datos que se muestren lo más reales posible, mediante lectores de flujo del tráfico. Por esto es tan importante la creación de un sistema de simulación de tráfico adecuado a nuestro problema.

6. Modelo

En este apartado describiremos el modelos que se va a usar para resolver el problema con lógica difusa.

En primer lugar la llegada de vehículos va a ser aleatoria. El proceso de selección estará dividido en 2 ciclos, el ciclo de luz roja y el ciclo de luz verde. El proceso asume que tienen una duración de 10 s/ciclo. Los vehículos dejan la cola en un ratio constante que llamaremos saturación que va a tener el valor de 3600 vehículos/h en ambos brazos del cruce.

Tenemos que si el vehículo llega como valor 1 y 0 si el vehículo no llega. Por lo que tenemos:

$$q_n = \begin{cases} 1, & \text{si el vehículo llega durante un intervalo } n \\ 0, & \text{si el vehículo no llega} \end{cases}$$

Tenemos que Q_G es el numero de vehículos que no han pasado en el estado de luz verde. Por lo que el número de vehículos totales en el intervalo n del modelo se da de la siguiente manera:

$$Q_n = Q_G + \sum_{i=1}^n q_i$$

El tiempo de espera de los vehículos viene dado por $D_{n,R}$ donde:

$$D_{n,R} = \sum_{i=1}^n (Q_G + \sum_{j=1}^i q_j)$$

Tenemos s como el flujo de saturación del cruce de los vehículos durante la fase de luz verde. Al comienzo de la fase de la luz verde tenemos que el número de vehículos no autorizados viene representado por S_n donde S_n es:

$$S_n = z * (Q_R + \sum_{i=1}^n q_i - s * n)$$

Donde Q_R es la cola que se formo durante el periodo rojo anterior y

$$z = \begin{cases} 1, & \text{Cuando multiplica por una cantidad no negativa} \\ 0, & \text{en otro caso} \end{cases}$$

Estos vehículos han sufrido un retraso de:

$$D_{n,G} = \sum_{i=1}^n z * (Q_R + \sum_{j=1}^i q_j - s * i)$$

Durante un ciclo el tiempo total de retraso viene dado por:

$$D = D_{R,R} + D_{G,G}$$

y la media de retraso por vehículo será:

$$d_m = \frac{D}{\sum_{i=1}^{R+G} q_n}$$

Esta media de retraso sera comparada con la media obtenida con la siguiente fórmula, la media anterior será la que use nuestro sistema:

$$d_t = \frac{C(1-\lambda)}{2(1-\lambda * X)} + \frac{X^2}{2_q(1-X)} - 0,65\left(\frac{C}{q^2}\right)^{\frac{1}{3}}X^{2+5\lambda}$$

Para nuestro estudio usamos la media calculada como d_m y mas tarde se comparará con el modelo que se construirá de lógica difusa. Una comparación de las dos medias calculadas se puede ver en la siguiente imagen.

N - S traffic (veh/hr)	E - W traffic (veh/hr)	Delay (secs/veh)		Error %
		Model (d_m)	Formula (d_t)	
360	360	7.2	7.4	- 2
360	720	7.4	7.9	- 6
360	1080	7.9	8.4	- 5
360	1440	8.4	9.0	- 7
360	1800	9.3	10.2	- 9
360	2160	12.3	12.9	- 5
360	2520	15.8	18.9	-17
720	720	9.7	10.0	- 3
720	1080	10.8	11.6	- 7
720	1440	12.7	13.8	- 8
720	1800	15.9	17.3	- 8
720	2160	21.8	24.9	-12
1080	1080	13.6	14.9	- 9
1080	1440	17.9	19.7	- 9
1080	1800	25.8	29.2	-11
1440	1440	27.3	30.7	-11

Figura 6.1: Tabla de comparación de medias calculadas.

En la tabla podemos ver la media calculada que es la que usa nuestro estudio y la media esperada que es la comparación, estas medias se usarán para comparar con el modelo de lógica difusa que veremos después.

7. Modelo de lógica difusa

El modelo de lógica difusa que se va a usar usa una interpretación cuantitativa, con 4 variables, estas variables son:

T =variable tiempo que se le asigna los valores: "very short", "short", "medium", "long", "very long".

A =Describe el número de vehículos que llegan a un brazo del cruce, puede tomar los valores: "none", "a few", "few", "many", "too many", "medium", "more than none", "more than a few", "more than few", "more than medium", "more than many", "more than too many".

Q =Describe los vehículos que hay en cola, puede tomar los valores: "any", "less than very small", "less than small", "less than small plus", "less than medium", "less than long", "less than very long".

E =Variable Extensión, tiene el mismo valor que T .

Para tomar decisiones se usarán instrucciones de la forma:

```
if T = medium and A = mt(medium) and Q = lt (small)
    then E = medium
else
    if T = long and A = mt(many) and Q = lt (medium)
        then E = long
```

Donde los operadores and y else son operadores de **mínimo** y **máximo**. En la frase P definida en $T \times A \times Q$,

if $T = \text{medium}$ and $A = \text{mt}(\text{medium})$ and $Q = \text{lt}(\text{small})$

Tenemos que $P = \min\{\mu_{\text{medium}}(T), \mu_{\text{mt}(\text{medium})}(A), \mu_{\text{lt}(\text{small})}(Q)\}$ Donde $\mu_{\text{medium}}(T)$ =grado de confianza de T , $\mu_{\text{mt}(\text{medium})}(A)$ = grado de confianza de A y $\mu_{\text{lt}(\text{small})}(Q)$ = grado de confianza de Q . Y $P = \mu_P(T, A, Q)$.

Tenemos:

```
if T= medium and A = mt(medium) and Q = lt (small)
    then E = medium
else
    if T= long and A = mt(many) and Q = lt (medium)
        then E = long
```

Donde la frase anterior formada por las instrucciones vista anteriormente se resuelve de la siguiente manera.

$$\max\{\min\{\mu_{\text{medium}}(T), \mu_{\text{mt}(\text{medium})}(A), \mu_{\text{lt}(\text{small})}(Q)\}, \min\{\mu_{\text{long}}(T), \mu_{\text{mt}(\text{many})}(A), \mu_{\text{lt}(\text{medium})}(Q)\}\}$$

Para obtener el grado de confianza de las distintas variables tenemos que mirar en las tablas que se muestran a continuación:

TABLE II FUZZY SETS DEFINED ON TIME (OR EXTENSION)										
time (secs) Fuzzy sets	1	2	3	4	5	6	7	8	9	10
very short	1	.5	0	0	0	0	0	0	0	0
short	0	.5	1	.5	0	0	0	0	0	0
medium	0	0	0	.5	1	.5	0	0	0	0
long	0	0	0	0	0	.5	1	.5	0	0
very long	0	0	0	0	0	0	0	.5	1	1

Figura 7.1: Tabla de tiempo y Extensión.

TABLE III FUZZY SETS DEFINED ON ARRIVALS										
Arrivals (veh.) F. sets	1	2	3	4	5	6	7	8	9	10
none	.5	.2	.1	0	0	0	0	0	0	0
a few	1	.5	.2	.1	0	0	0	0	0	0
few	.5	1	.5	.2	.1	0	0	0	0	0
medium	.2	.5	1	.5	.2	.1	0	0	0	0
many	.1	.2	.5	1	.5	.2	.1	0	0	0
too many	0	.1	.2	.5	1	.5	.2	.1	0	0

Figura 7.2: Tabla de coches que llegan.

TABLE IV
FUZZY SETS DEFINED ON QUEUES

queue (veh)	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32			
f.sets																																
very small	0	.5	.7	.9	.1	.9	.7	.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
small	0	0	0	0	0	.5	.7	.9	1	.9	.7	.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
small plus	0	0	0	0	0	0	0	0	0	.5	.7	.9	1	.9	.7	.5	0	0	0	0	0	0	0	0	0	0	0	0	0			
medium	0	0	0	0	0	0	0	0	0	0	0	0	0	.5	.7	.9	1	.9	.7	.5	0	0	0	0	0	0	0	0	0			
long	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.5	.7	.9	1	.9	.7	.5	0	0	0	0	0			
very long	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.5	.7	.9	1	.9	.7	.5	0			

Figura 7.3: Tabla de coches en cola.

TABLE V
"MORE THAN"-OPERATION ON FUZZY SETS OF TABLE III

Arrivals (vehicles)	1	2	3	4	5	6	7	8	9	10
fuzzy sets										
mt(none)	.5	.8	.9	1	1	1	1	1	1	1
mt(a few)	0	.5	.8	.9	1	1	1	1	1	1
mt(few)	0	0	.5	.8	.9	1	1	1	1	1
mt(medium)	0	0	0	.5	.8	.9	1	1	1	1
mt(many)	0	0	0	0	.5	.8	.9	1	1	1
mt(too many)	0	0	0	0	0	.5	.8	.9	1	1

Figura 7.4: Tabla de more than.

TABLE VI
"LESS THAN"-OPERATION ON FUZZY SETS OF TABLE IV

queue (veh)	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
f. sets																															
lt(very small)	1	.5	.3	.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
lt(small)	1	1	1	1	1	.5	.3	.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
lt(small plus)	1	1	1	1	1	1	1	1	1.5	.3	.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
lt(medium)	1	1	1	1	1	1	1	1	1	1	1	1	1.5	.3	.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
lt(long)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.5	.3	.1	0	0	0	0	0	0	0	0	0	0		
lt(very long)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.5	.3	.1	0	0	0	0	0	0		

Figura 7.5: Tabla de less than.

La descripción del proceso para decidir las acciones que se aplican vienen dadas como sigue.

Consideramos como el brazo correcto el brazo N-S del cruce y hay en cola por ejemplo 5 coches en el brazo E-O. Si tomamos un tiempo de 10s para el rango de tiempo de acción, creamos un vector con 10 posiciones cada una perteneciente a un segundo del tiempo dentro del rango de os 10s. en cada segundo tenemos el numero de vehículos que nos llega al brazo de cruce N-S. Por lo que nos queda un vector:

$$(0, 1, 0, 1, 1, 1, 1, 0, 0, 1)$$

Para el brazo E-O tenemos que:

$$(0, 1, 0, 0, 1, 0, 0, 1, 0, 0)$$

Desde estos vectores construimos otros donde ponemos las suma acumulada en cada segundo de los coches que han llegado en el instante de tiempo correspondiente, por lo que tenemos que para N-S:

$$(0, 1, 1, 2, 3, 4, 5, 5, 5, 6)$$

Y para el brazo E-O añadimos a la suma acumulativa en cada segundo los coches que tenemos en cola para ese brazo que eran 5, entonces tenemos que:

$$(5, 6, 6, 6, 7, 7, 7, 8, 8, 8)$$

Podemos ver entonces que para el segundo 1 del brazo N-S no llega ningún coche y para el brazo E-O tenemos 5 que son los que hay en cola.

Para el segundo 4 tenemos que para el brazo N-S han llegado hasta ese momento 2 coches y para el brazo E-O tenemos que hay 6 coches uno que ha llegado nuevo hasta ese momento y los 5 que había en cola.

Con estos datos podemos empezar a poner valores a nuestras variables para ver el grado de confianza de estos valores y poder resolver nuestras reglas.

Aplicaremos una regla para cada instante de tiempo de los vectores visto antes y aplicando los valores que se ajusten a nuestro estudio, en este caso vamos a poner unos valores de ejemplo . De forma que para el segundo 1 tenemos que:

$$t = 1, a = 0, q = 5, e = 1$$

Donde: $(t, a, q, e) = (1, 0, 5, 1) = \min\{\mu_{veryshort}(1), \mu_{mt(none)}(a), \mu_{any}(q)\}$ el grado de confianza de e es igual que el de t por lo que no lo ponemos, valor de any se coge el mínimo valor de todos los valores posibles para la columna 5 en la tabla queue. Por lo que tenemos que:

$$(1, 0, 5, 1) = \min\{1, 0, 5, 0\}$$

Para conseguir estos valores miramos en sus tablas respectivas, para el valor de t miramos en la tabla e tiempo, para el valor de a miramos en la tabla de coches que llegan y para el valor de q miramos en la tabla de coches en cola. Los valores $t=1$, $a=0$, $q=5$ son el número de la columna que hay que mirar en cada tabla, $t=$ igual al segundo correspondiente, $a=a$ los coches llegados hasta ese segundo para el brazo N-S, $q=$ Los coches en cola para el brazo E-O. Las filas vienen dados por los valores "very short" para t, "mt(none)" para a, y ".any" para q.

Los valores encontrados en las tablas son los grados de confianza, después se selecciona el mínimo de los grados de confianza.

Para este ejemplo el mínimo del grado de confianza es 0.

En este estudio se usaron 5 reglas en 5 intervalos de tiempo distintos, las reglas que se usaron fueron:

Intervention: 7th second

```
if      T = very short
  and   A = mt(none)
  and   Q = any
then    E = very short
else
  if      T = short
    and   A = mt(a few)
    and   Q = lt(very small)
  then    E = short
else
  if      T = medium
    and   A = mt(few)
    and   Q = lt(very small)
  then    E = medium
else
  if      T = long
    and   A = mt(medium)
    and   Q = lt(very small)
  then    E = long
else
  if      T = very long
    and   A = mt(many)
    and   Q = lt(very small)
  then    E = very long.
```

Figura 7.6: Reglas 1.

Intervention: 17th second

```
if      T = very short
  and   A = mt(none)
  and   Q = any
then    E = very short
else
  if      T = short
    and   A = mt(a few)
    and   Q = lt(very small)
  then    E = short
else
  if      T = medium
    and   A = mt(few)
    and   Q = lt(very small)
  then    E = medium
else
  if      T = long
    and   A = mt(medium)
    and   Q = lt(very small)
  then    E = long
else
  if      T = very long
    and   A = mt(many)
    and   Q = lt(small)
  then    E = very long.
```

Figura 7.7: Reglas 2.

Intervention: 27th second

```
if      T = very short
  and   A = mt(none)
  and   Q = any
then    E = very short
else
  if     T = short
    and  A = mt(a few)
    and  Q = lt(very small)
  then   E = short
else
  if     T = medium
    and  A = mt(few)
    and  Q = lt(very small)
  then   E = medium
else
  if     T = long
    and  A = mt(medium)
    and  Q = lt(very small)
  then   E = long
else
  if     T = very long
    and  A = mt(many)
    and  Q = lt(small)
  then   E = very long.
```

Figura 7.8: Reglas 3.

Intervention: 37th second

```
if      T = very short
  and   A = mt(none)
  and   Q = any
then    E = very short
```

Figura 7.9: Reglas 4.

```
else
  if     T = short
    and  A = mt(a few)
    and  Q = lt(small plus)
  then   E = short
else
  if     T = medium
    and  A = mt(medium)
    and  Q = lt(small plus)
  then   E = medium
else
  if     T = long
    and  A = mt(many)
    and  Q = lt(medium)
  then   E = long
else
  if     T = very long
    and  A = mt(too many)
    and  Q = lt(long)
  then   E = very long.
```

Figura 7.10: Reglas 5.

```

Intervention: 47th second
    if       $T = \text{very short}$ 
      and    $A = mt(\text{none})$ 
      and    $Q = \text{any}$ 
    then     $E = \text{very short}$ 
  else
    if       $T = \text{short}$ 
      and    $A = mt(\text{a few})$ 
      and    $Q = lt(\text{long})$ 
    then     $E = \text{short}$ 
  else
    if       $T = \text{medium}$ 
      and    $A = mt(\text{medium})$ 
      and    $Q = lt(\text{long})$ 
    then     $E = \text{medium}$ 
  else
    if       $T = \text{long}$ 
      and    $A = mt(\text{too many})$ 
      and    $Q = lt(\text{very long})$ 
    then     $E = \text{long}$ 
  else
    if       $T = \text{very long}$ 
      and    $A = mt(\text{too many})$ 
      and    $Q = lt(\text{very long})$ 
    then     $E = \text{very long.}$ 

```

Figura 7.11: Reglas 6.

Con estas reglas y usando el modelo de lógica difusa visto en este apartado, se ha comparado los resultados obtenidos con el modelo visto en el apartado anterior y los resultados fueron:

TABLE VIII
COMPARISON BETWEEN DELAYS CAUSED BY EFFICIENT VEHICLE-ACTUATED
CONTROLLER AND FUZZY LOGIC ONE

N - S traffic (veh/hr)	E - W traffic (veh/hr)	Average overall delay (secs/veh)		Improvement %
		Vehicle-actuated controller	Fuzzy-logic controller	
360	360	7.2	5.7	+21
360	720	7.4	6.1	+18
360	1080	7.9	6.6	+17
360	1440	8.4	7.3	+13
360	1800	9.3	8.4	+10
360	2160	12.3	10.0	+19
360	2520	15.8	13.6	+14
720	720	9.7	7.4	+21
720	1080	10.8	8.8	+19
720	1440	12.7	10.9	+14
720	1800	15.9	14.1	+11
720	2160	21.8	18.5	+15
1080	1080	13.6	12.0	+12
1080	1440	17.9	15.4	+14
1080	1800	25.8	21.6	+16
1440	1440	27.3	22.9	+16

Figura 7.12: Resultados.

Podemos ver que en los resultados obtenidos se obtuvieron mejoras sustanciales al usar el controlador de lógica difusa. Aunque los resultados del controlador de lógica difusa fueron mejores que los del modelo del controlador anterior, en el estudio del modelo de lógica difusa, se han probado las reglas por ensayo y error por lo que se pueden mejorar los resultados obtenidos mediante un controlador de lógica difusa.

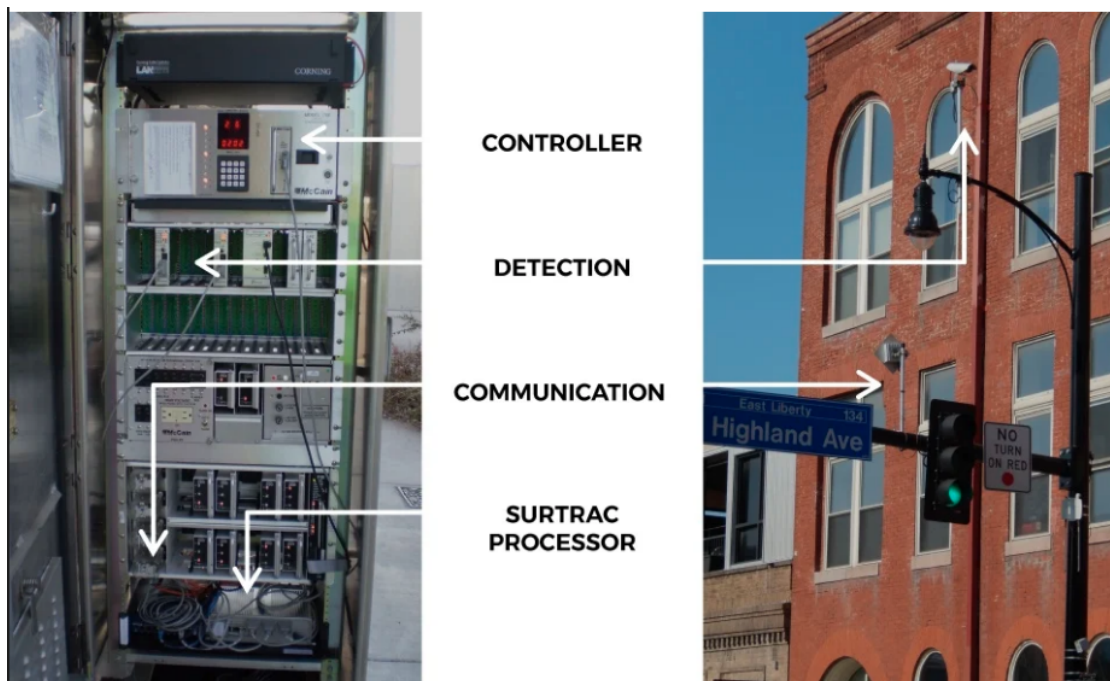
8. Uso de los semáforos inteligentes alrededor del mundo

Está claro que hay decenas y decenas de papers sobre el tema que tratamos, pero, ¿se han llegado a usar estas técnicas en casos reales?

8.1. Pittsburg, Pennsylvania (USA)

”Todos conocemos los atascos. Causan estrés, pérdida de dinero y tiempo y una emisión extra de dióxido de carbono”. Así empieza artículo de Smart City Hub [11]. En el, se muestra el caso de Pittsburg, una ciudad de Pennsylvania, situada en Estados Unidos.

Dicha ciudad ha conseguido reducir este terrible problema en un 40 %, debido, entre otras cosas, a la ayuda de los semáforos inteligentes. Como podemos ver, usan la tecnología Surtrac [12]. Indagando un poco más en la implementación de esta tecnología [13], la compañía Rapid Flow usa tanto controladores de semáforos como cámaras para la detección de los vehículos, como podemos ver en la siguiente imagen:



Para ser más exactos, no he encontrado referencia a la utilización de lógica difusa, pero tampoco a la no utilización de la misma. Al fin y al cabo es un producto de esta compañía, y le conviene no enseñar todas sus cartas a la competencia. Lo más probable es que usen lógica difusa para el comportamiento de sus controladores.

8.2. Artículo en MESinsights

Podemos encontrar actualmente, que el los semáforos inteligentes están a la orden del día. Podemos ver como Cate Lawrence y Nicole Kareta hablan en su artículo [14] (2020) de cómo puede ayudar el uso de estos a las poblaciones, reduciendo los tiempos de espera de conductores y peatones.

8.3. NoTraffic

Vemos otra empresa estadounidense [15, 16], que ha realizado estudios piloto en varias ciudades de Estados Unidos. Reportan que en sus pruebas, la tecnología redujo el tiempo de espera en colas en 2700 horas, mientras que reducen las emisiones en 33 toneladas en cada cruce por año. Actualmente, se han asociado con Phoenix, Arizona, para realizar otras pruebas, y es posible que en el futuro se haga un despliegue mucho más amplio de sus servicios.

Esta empresa espera que en el futuro se pueda afrontar el problema de los más de 300000 semáforos que hay en Estados Unidos, aunque claro, en ningún momento comentan acerca del precio de su sistema, por lo que aún queda mucho para poder hacer frente al problema de manera real.

Referencias

- [1] Cihan Karakuzu, Osman Demirci. *Fuzzy logic based smart traffic light simulator design and hardware implementation*. Applied Soft Computing 10, 2010
- [2] C. Pappis, E. Mamdani. *A fuzzy logic controller for a traffic junction*. IEEE Transactions on Systems, Man, and Cybernetics, 1977.
- [3] M. Nakatsuyama, H. Nagahashi, N. Nishizuka. *Fuzzy logic phase controller for traffic functions in the one-way arterial road*. Proc. IFAC 9th Triennial World Congress, Pergamon Press, Oxford, 1984.
- [4] J. Favilla, A. Machion, F. Gomide. *Fuzzy traffic control: adaptive strategies*. Second IEEE International Conference on Fuzzy Systems II, 1993.
- [5] M.B. Trabia, M.S. Kaseko, M. Ande. *A two-stage fuzzy logic controller for traffic signals*. Transportation Research: Part C 7, 1999.
- [6] Ella Bingham. *Reinforcement learning in neurofuzzy traffic signal control*. European Journal of Operational Research 131, 2001.
- [7] C.-H. Chou, J.-C. Teng. *A fuzzy logic controller for traffic junction signals*. Information Sciences 143, 2002.
- [8] Y. Chong, C. Quek, P. Loh. *A novel neuro-cognitive approach to modeling traffic control and flow based on fuzzy neural techniques*. Expert Systems with Applications 36 (3 Part1), 2009.
- [9] J.-D. Schmöcker, S. Ahuja, M.G.H. Bell. *Multi-objective signal control of urban junctions-Framework and a London case study*. Transportation Research Part C: Emerging Technologies 16, 2008.
- [10] Maja Kalinic, Jukka M. Krispl. *Fuzzy inference approach in traffic congestion detection*. Annals of GIS, 25:4, 2019.
- [11] Smart City Hub. <http://smartcityhub.com/mobility/smart-traffic-control/>. 2017.
- [12] Surtrac. <https://www.rapidflowtech.com/surtrac>.
- [13] Implementación del algoritmo de Surtrac. <https://www.rapidflowtech.com/surtrac/implementation>.
- [14] MESinsights. <https://www.mes-insights.com/smart-city-infrastructure-improves-the-journey-of-road-users-a-959373/>. 2020.
- [15] Kristin Houser . <https://www.freethink.com/articles/smart-traffic-lights>. 2020.
- [16] NoTraffic. <https://notraffic.tech/>.