



**Facultad de Ciencias Naturales y Matemáticas**

**Sincronización de semáforos basado en el modelo  
MAXBAND y su relación con base de ciclos**

**Proyecto Integrador**

Previo a la obtención del título de:

**Matemático**

Presentado por:

**Jhixon Macías  
Juan José Costa**

ESPOL  
Guayaquil-Ecuador  
2022

# Dedicatoria

A mi amada Alenka Calderón Morán.

*Jhixon Macías*

A todas las personas que siempre han estado ahí y me han ayudado a ser la persona que soy.

*Juan José Costa*

# Agradecimientos

A Dios,  
a mis queridos padres,  
a mi bienquisto Team,  
a mi amigo JuanJo,  
al Dr. Elvis Aponte,  
al Dr. Xavier Cabezas,  
a todos ellos por su apoyo y compañía a  
lo largo de mi pregrado.

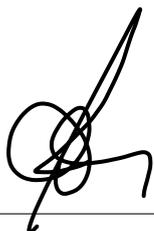
*Jhixon Macías*

A Dios y a mi Madre del Cielo,  
a mi familia y padres por su apoyo,  
a mi gran amigo Jhixon,  
a Mel, Cris, Marco y Alexander,  
al Dr. Xavier Cabezas por la ayuda  
brindada.

*Juan José Costa*

# Declaración

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Jhixon Steven Macías García y Juan José Costa Fassio damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



---

Jhixon Steven Macías García



---

Juan José Costa Fassio

# Evaluadores

---

Luz Elimar Marchan M. Ph.D.  
PROFESORA DE LA  
MATERIA

---

Xavier Cabezas G. Ph.D.  
PROFESOR TUTOR

# Resumen

La sincronización deficiente de semáforos ha representado un problema notable desde su implementación alrededor del mundo. Sus efectos van desde pérdidas económicas, daños ambientales y complicaciones en la salud de las personas. Por esta razón, se propone optimizar la sincronización de semáforos a través de la formulación MAXBAND. Esta formulación permite obtener una sincronización tal que maximiza el tiempo en el cual un carro, a determinada velocidad, puede pasar por las luces verdes de un número determinado de semáforos sin parar (bandwith). En este trabajo se utilizó el lenguaje de programación Python junto con el paquete de programación lineal Pyomo y el solver CPLEX para la implementación de la formulación MAXBAND como un problema de programación lineal. Se empezó programando la formulación más básica del caso arterial hasta llegar a la formulación más general con restricciones de ciclos sobre una red, en cuyo caso también se utilizó Wolfram Mathematica para encontrar las bases de ciclos necesarias. Considerando un conjunto de calles de un sector del norte de Guayaquil y un tiempo de ciclo para las luces verdes y rojas entre 55 y 75 segundos por semáforo y límites de velocidad entre 13.4 m/s y 17.9 m/s; se obtuvo, para las diferentes calles consideradas, un tiempo óptimo de bandwith que oscila entre 0.52 y 0.58 ciclos. Finalmente, este trabajo puede aplicarse en cualquier ciudad del mundo para la sincronización eficiente de semáforos, sin la necesidad de recurrir a costosos programas que hagan el mismo trabajo.

**Plabras claves:** Sincornización de semáforos, formulación MAXBAND, bandwith, programación lineal.

# Abstract

Bad traffic light synchronization has always been a problem since its implementation. The impact goes from economic losses and environmental harm to health problems. Consequently, traffic light synchronization through MAXBAND formulation is proposed. This formulation obtains a maximized synchronization of the time in which a car, at a certain velocity, can pass through a determined number of green lights without stopping (bandwidth). In this work, the Python programming language was used along with the linear programming package Pyomo and CPLEX solver, for the implementation of the MAXBAND formulation as a linear programming problem. The programming started with the most basic formulation of the arterial case until reaching the general formulation with cycle constraints over a network, in which case Wolfram Mathematica was used to find the necessary cycle basis. Considering a set of streets in the north of Guayaquil, a cycle time for green lights and red lights between 55 and 75 seconds per signal, and a speed limit between 13.4 m/s and 17.9 m/s, for the chosen streets, the optimum bandwidth time was between 0.52 and 0.58 cycles. Finally, this work can be applied in any city in the world for efficient signal synchronization, without needing to resort to expensive programs that do the same job.

**Key Words:** Traffic light synchronization, MAXBAND formulation, Bandwidth, Linear Programming.

# Índice de figuras

1.1.	Diagrama Geométrico MAXBAND para una arteria. . . . .	4
1.2.	Patrones de fases de giro a la izquierda. . . . .	7
1.3.	El ciclo $\zeta_e$ en $T \cup e$ . . . . .	9
1.4.	Base de ciclo débilmente fundamental formada por los seis triángulos visibles. . . . .	10
1.5.	Base de ciclo formada con los cuatro triángulos visibles. . . . .	10
1.6.	Grid graph $G_{3 \times 4}(E, V)$ . . . . .	11
1.7.	Unión $i$ de las arterias principal ( $m$ ) y transversal ( $c$ ) . . . . .	13
1.8.	Geometría de $\Psi_{mc}^{p_m p_c}$ . . . . .	13
2.1.	Diagrama espacio-tiempo donde se ilustra la sincronización de semáforos óptima en función de $\theta_{ij}$ . . . . .	18
2.2.	Ejemplo de Red con 7 señales con 5 arterias. . . . .	21
2.3.	Base de ciclo fundamental conformada por dos ciclos. El ciclo 1: (7, 4, 5, 6) y el ciclo 2: (7, 4, 3, 2, 1). . . . .	22
2.4.	Un sector del norte de Guayaquil y Grafo que abstrae las arterias y nodos de las calles . . . . .	23
2.5.	Base de ciclo conformada por dos ciclos. El ciclo 1: (9, 5, 6, 7, 8) y el ciclo 2: (1, 2, 4, 5, 6, 7, 8, 9). . . . .	24
2.6.	Base de ciclo conformada por dos ciclos. El ciclo 1: (9, 5, 6, 7, 8) y el ciclo 2: (1, 2, 4, 5, 9). . . . .	25
3.1.	Representación Bandwidth y tiempos de luces rojas y verdes del out-bound e inbound . . . . .	27
3.2.	Diagrama espacio-tiempo de sincronización para el caso arterial 1. . . . .	28
3.3.	Diagrama espacio-tiempo de sincronización para el caso arterial 2. . . . .	29
3.4.	Diagrama espacio-tiempo de sincronización para el caso arterial 3. . . . .	30
3.5.	Diagrama espacio-tiempo de sincronización para el caso sobre una red: arteria 1. . . . .	34
3.6.	Diagrama espacio-tiempo de sincronización para el caso sobre una red: arteria 2. . . . .	34
3.7.	Diagrama espacio-tiempo de sincronización para el caso sobre una red: arteria 3. . . . .	35
3.8.	Diagrama espacio-tiempo de sincronización para el caso sobre una red: arteria 4. . . . .	35

3.9. Diagrama espacio-tiempo de sincronización para el caso sobre una red: arteria 5. . . . .	36
3.10. Diagrama espacio-tiempo de sincronización para el caso real: arteria 1.	38
3.11. Diagrama espacio-tiempo de sincronización para el caso real: arteria 2.	39
3.12. Diagrama espacio-tiempo de sincronización para el caso real: arteria 3.	39
3.13. Diagrama espacio-tiempo de sincronización para el caso real: arteria 4.	40
3.14. Diagrama espacio-tiempo de sincronización para el caso real: arteria 5.	40

# Índice de cuadros

1.1. Expresiones para $\Delta_i$ en función de los cuatros patrones de giros hacia la izquierda. . . . .	7
2.1. Arterias etiquetadas en función de los nodos. . . . .	22
2.2. Duración en ciclos de las luces rojas por arteria. . . . .	22
2.3. Arterias etiquetadas en función de los nodos. . . . .	26
2.4. Duración en ciclos de las luces rojas por arteria. . . . .	26
3.1. Valores de $\theta(1, h)$ para $h$ . Caso Arterial 1 . . . . .	28
3.2. Valores de $w_i$ para cada señal $i$ , con $w_i = \bar{w}_i$ para todo $i$ . Caso arterial 1	28
3.3. Valores de $\theta(1, h)$ para $h$ . Caso arterial 2 . . . . .	29
3.4. Valores de $w_i$ para cada señal. Con $w_i = \bar{w}_i$ para todo $i$ . Caso arterial 2	30
3.5. Valores obtenidos de las velocidades entre señales a los que se debe circular para aprovechar el bandwith. Caso arterial 2 . . . . .	30
3.6. Valores de $\theta(1, h)$ para $h$ . Caso arterial 3 . . . . .	31
3.7. Valores de $w_i$ para cada señal $i$ . Caso Arterial 3 . . . . .	31
3.8. Valores obtenidos de las velocidades entre señales a los que se debe circular para aprovechar el bandwith. Caso arterial 3. . . . .	31
3.9. Fases de giro a la izquierda para cada señal $i$ , en función de variables binarias $\delta_i$ y $\bar{\delta}_i$ . Caso arterial 3 . . . . .	31
3.10. Valores de $\theta(1, h)$ y $\bar{\theta}(1, h)$ para cada arteria. Caso sobre una red .	32
3.11. Valores de $w_i$ y $\bar{w}_i$ para cada arteria. Caso sobre una red. . . . .	32
3.12. Valores de $v_{i,i+1}$ y $\bar{v}_{i,i+1}$ para cada arteria. Caso sobre una red . . .	32
3.13. Fases de giro a la izquierda para cada señal de cada arteria, en función de las variables binarias $\delta_i$ y $\bar{\delta}_i$ . Caso general sobre una red . . . . .	33
3.14. Valores del bandwith para cada arteria en dirección de entrada y salida. Caso general sobre una red. . . . .	33
3.15. Valores de $\theta(1, h)$ y $\bar{\theta}(1, h)$ para cada arteria. Caso real. . . . .	36
3.16. Valores de $w_i$ y $\bar{w}_i$ para cada arteria. Caso real . . . . .	37
3.17. Valores de $v_{i,i+1}$ y $\bar{v}_{i,i+1}$ para cada arteria. Caso real . . . . .	37
3.18. Fases de giro a la izquierda para cada señal, en función de las variables $\delta_i$ y $\bar{\delta}_i$ . Caso Real . . . . .	37
3.19. Valores del bandwith para cada arteria en dirección de entrada y salida. Caso Real . . . . .	38
3.20. Tiempo de ejecución del modelo lineal MAXBAND. . . . .	41

# Índice general

<b>Resumen</b>	<b>VI</b>
<b>Abstract</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del problema . . . . .	1
1.2. Justificación del problema . . . . .	1
1.3. Objetivos . . . . .	2
1.3.1. Objetivo General . . . . .	2
1.3.2. Objetivos Específicos . . . . .	2
1.4. Marco teórico . . . . .	2
1.4.1. Introducción y estado del arte . . . . .	2
1.4.2. Formulación MAXBAND: caso arterial . . . . .	3
1.4.3. Ciclos y base de ciclos . . . . .	8
1.4.4. Formulación MAXBAND sobre una red . . . . .	11
1.4.5. Una breve introducción a la teoría de Webster . . . . .	14
<b>2. Metodología</b>	<b>16</b>
2.1. Sobre la sincronización de semáforos: Formulación MAXBAND . . . . .	17
2.2. Modelo lineal: caso arterial 1 . . . . .	19
2.3. Modelo lineal: caso arterial 2 . . . . .	19
2.4. Modelo lineal: caso arterial 3 . . . . .	20
2.5. Modelo lineal: caso general sobre una red . . . . .	21
2.6. Modelo lineal: caso real . . . . .	23
<b>3. Resultados de las simulaciones</b>	<b>27</b>
3.1. Sobre los diagramas espacio-tiempo . . . . .	27
3.2. Resultado del modelo lineal: caso arterial 1 . . . . .	28
3.3. Resultado del modelo lineal: caso arterial 2 . . . . .	29
3.4. Resultado del modelo lineal: caso arterial 3 . . . . .	30
3.5. Resultado del modelo lineal: caso sobre una red . . . . .	32
3.6. Resultado del modelo lineal: caso real . . . . .	36
<b>4. Conclusiones y recomendaciones</b>	<b>42</b>
4.1. Conclusiones . . . . .	42
4.2. Recomendaciones . . . . .	43

<b>A. Código Python: Caso sobre una red</b>	<b>46</b>
<b>B. Código Grafo y Base de Ciclos Wolfram Mathematica</b>	<b>58</b>

# Capítulo 1

## Introducción

En el presente capítulo se introduce el problema de sincronización de semáforos, se expone una breve revisión del estado del arte respecto a los modelos matemáticos asociados a la sincronización de semáforos, se presenta la formulación MAXBAND para una arteria y una red general, así como también una breve introducción a la teoría de ciclos y base de ciclos sobre un grafo, y a la teoría de Webster.

### 1.1. Descripción del problema

Los semáforos, también conocidos como señales de control de tráfico, son dispositivos de señalización que se sitúan en intersecciones viales cuyo funcionamiento se remonta al siglo XIX y consiste en regular el tráfico de vehículos y peatones dentro de zonas urbanas alrededor del mundo.

La sincronización deficiente de semáforos incrementa el riesgo de accidentes de tránsito y aumenta el índice de tráfico en las ciudades. Un índice de tráfico elevado, que se traduce en congestión vehicular, implica un aumento de la emisión de gases de efecto invernadero provenientes de vehículos motorizados, que de hecho son, hoy en día, su principal fuente de emisión. La presencia excesiva de gases de efecto invernadero, como es bien sabido, trae consecuencias negativas al medio ambiente y a la salud de las personas. Por otro lado, la congestión vehicular también implica un aumento de los niveles de estrés individual y colectivo de los ciudadanos que utilizan diferentes medios de transporte en medio de un tráfico congestionado, deteriorando progresivamente su salud física y mental. Más aún, afecta a varias empresas e instituciones cuyo sector productivo depende de medios de transporte.

Por lo anteriormente expuesto, podemos concluir que la sincronización deficiente de semáforos es un problema cuyo alcance está al nivel de varias ciudades alrededor del mundo.

### 1.2. Justificación del problema

Solo para el 2021, entre las primeras 10 ciudades del mundo con mayor índice de tráfico, según el *Global Traffic Scorecard INREX (2021)*, se encontraron: London,

París, Brussels, Moscow, New York City NY, Chicago IL, Roma, Bogotá, Palermo e Istanbul. El promedio de horas perdidas durante este año, en horas pico, fue de 113. En cuanto a pérdida de dinero en dólares por congestión vehicular, en países como E.E.U.U, Inglaterra y Alemania, esta llega a 53 billones, 83 billones y 31 billones, respectivamente. Es claro que, por lo mencionado anteriormente, la congestión vehicular es un serio problema de vanguardia alrededor del mundo y, la principal causa de este es la sincronización deficiente de semáforos.

Bajo este contexto, en el presente proyecto se implementará la formulación MAXBAND, propuesta por Little et al. (1981) y estudiada recientemente por Cabezas et al. (2019), en Pyomo-Python con el fin de simular experimentalmente (con datos reales), una sincronización de semáforos en ciertos sectores de alguna ciudad, misma que se espera sea más eficiente que la actual. Cabe mencionar que la formulación MAXBAND, a diferencia de otras formulaciones, se basa en una función objetivo y restricciones lineales. Esto supone, desde luego, un mejor desempeño computacional en cuanto a conseguir resultados eficientes y prácticos en una menor cantidad de tiempo.

## 1.3. Objetivos

### 1.3.1. Objetivo General

Mejorar la sincronización de semáforos en algunos sectores de alguna ciudad mediante el uso de un modelo matemático basado en MAXBAND y su relación con base de ciclos para disminuir medidas de desempeño del tráfico.

### 1.3.2. Objetivos Específicos

- Adaptar la formulación MAXBAND en Pyomo-Python para su implementación en simulaciones experimentales de sincronización de semáforos;
- Evaluar el impacto de la Base de Ciclos sobre el grafo que representa una red de transporte en la formulación MAXBAND;
- Comparar el desempeño del tráfico obtenido de forma experimental a partir de la sincronización de semáforos MAXBAND con el desempeño del tráfico actual en algunos sectores de alguna ciudad.

## 1.4. Marco teórico

### 1.4.1. Introducción y estado del arte

En la actualidad, la congestión del tráfico se ha convertido en un tema de gran impacto debido al aumento de vehículos en las vías de las zonas urbanas alrededor del mundo. Para controlar el flujo de vehículos, los semáforos se han utilizado como

agentes reguladores desde el siglo XIX. En este sentido, la sincronización de semáforos es un tema de notable importancia, el cual se aborda comúnmente siguiendo dos enfoques, minimizando las demoras o algunas otras medidas que evalúan el rendimiento del tráfico, o maximizando el tiempo en que los vehículos se mueven sin detenerse en las luces rojas, también conocido como la maximización del ancho de banda (Bandwidth), véase (Papageorgiou et al., 2003).

En cuanto al flujo del tráfico, por lo general, se formula un problema de minimización para reducir el retraso general o el número de paradas completas de los vehículos. En un estudio clásico, Gartner et al. (1975) propusieron una formulación no lineal convexa con una linealización por partes y un mayor número de restricciones. De manera similar, un modelo lineal análogo permitió decidir entre diferentes planes de temporización de señales mientras se relajaba la suposición de un ancho de período rojo-verde común, véase (Wünsch, 2008).

Con respecto a la maximización del ancho de banda, las formulaciones iniciales modelaban un método geométrico para una calle de dos sentidos dado un período de tiempo rojo-verde fijo y común en cada señal, donde se asumía que los vehículos se conducían a una velocidad preasignada (constante). Posteriormente, se propuso un programa lineal entero mixto (MILP) para resolver una versión más completa del problema, véase (Morgan et al., 1964) y (Little, 1966), respectivamente.

Más tarde, se incluyeron variaciones de velocidad y límites en los límites de velocidad entre señales adyacentes. Además, se proporcionó una extensión para resolver el mismo problema en redes generales, pero solo se pudieron resolver instancias muy pequeñas dado que es un problema más difícil, de hecho es un problema **NP**-duro, ya que es necesario introducir las llamadas restricciones de bucle, que permiten modelar los movimientos circulares que pueden realizar los vehículos, lo cual condujo al modelo MAXBAND, introducido por Little et al. (1981).

Respecto a la resolución para la sincronización de semáforos en redes de transporte basado en MAXBAND, Cabezas et al. (2019) implementaron una metaheurística híbrida que combina las heurísticas Tabu Search y Variable Neighborhood Search. Además de esto, incorporaron algunos ajustes a la formulación MAXBAND, como la revisión de las restricciones del modelo lineal entero mixto, incluidas las que describen todas las rutas cíclicas en un grafo; y generalizaron algunos límites para las variables enteras, adaptando el modelo para estudiar situaciones más realistas.

### 1.4.2. Formulación MAXBAND: caso arterial

En general, como ya se ha mencionado, el objetivo de la formulación MAXBAND es sincronizar un conjunto de semáforos (sobre una arteria o red) de tal forma que el bandwidth de esta, sea el máximo. Formalmente:

**Definición 1.** (Morgan et al., 1964, Bandwidth) *Considere una arteria sobre una red de transporte con una sucesión de semáforos ubicados sobre sus esquinas. El bandwidth (ancho de banda) a lo largo de la arteria, es la porción del periodo de tiempo durante el cual un vehículo podrá comenzar en un esquina, y por medio de viajar a una velocidad preasignada, ir a otra sin parar por una luz roja.*



- $w_i(\bar{w}_i)$  : Tiempo desde el lado derecho (izquierdo) de la luz roja de la señal  $S_i$  hasta el lado izquierdo (derecho) del bandwidth verde en dirección de salida (entrada), en periodos.
- $\Delta_i$  : Tiempo desde el centro de  $\bar{r}_i$  hasta el centro más cercano de  $r_i$ , en periodos. Es positivo de izquierda a derecha.

**Observación 1.** *El tamaño del bandwidth (entrada o salida) debe estar limitado por el tamaño de la luz verde (entrada o salida), véanse las ecuaciones (1.8) y (1.9) del modelo lineal 1.4.1. Por otro lado, los parámetros  $\tau_i$  y  $\bar{\tau}_i$  permiten que un grupo finito de autos (no necesariamente uno) se beneficie del bandwidth. Las variables  $w_i$  y  $\bar{w}_i$  indican el inicio del bandwidth dentro del periodo de una luz verde. Finalmente, la variable  $\Delta_i$  permite considerar patrones de giros hacia la izquierda.*

Para la formulación, se supone que todas las señales funcionan en un período  $T$  de señal común cuya duración se introduce en el modelo como una variable de decisión. Esta señal debe estar en un intervalo  $[T1, T2]$ , por supuesto, la variable de decisión  $z$  es el recíproco de la duración del período, es decir,  $z = 1/T$ . Véase la ecuación (1.7) del modelo lineal 1.4.1.

Por otro lado, se supone que dentro de la arteria los vehículos (de una señal a otra  $S_i$  a  $S_j$  ya sea de salida o entrada) viajan a una determinada velocidad, digamos  $v_i$  (en metros sobre segundos) y ésta resulta ser una variable de decisión acotada por un límite inferior y superior, sean éstos  $e_i$  y  $f_i$ , respectivamente. Además, se denota por  $d_i$  (en metros) a la distancia entre la señal  $S_i$  y  $S_j$ .

Ahora bien, para evitar cambios repentinos de las velocidades entre señales consecutivas, estas se limitan imponiendo una cota superior e inferior, digamos  $1/h_i$  y  $1/g_i$  sobre los recíprocos de las velocidades. La razón de utilizar restricciones recíprocas sobre la velocidades y el cambio de velocidad, es que permiten linealizar estas mismas restricciones sobre el modelo. Primero note que no es posible considerar directamente las desigualdades  $e_i \leq v_i \leq f_i$  porque la duración del período también es una variable. Así, si  $v_i$  se transforma de metros sobre segundo a metros sobre periodo, la desigualdad se convierte en  $e_i T \leq v_i \leq f_i T$ , misma que no es lineal. Pero, usando los recíprocos de los límites de las velocidades, se tiene que:

$$e_i \leq v_i \leq f_i \rightarrow \frac{d_i}{f_i} \leq \frac{d_i}{v_i} \leq \frac{d_i}{e_i} \rightarrow \frac{d_i}{f_i} z \leq t_i \leq \frac{d_i}{e_i} z,$$

y usando los recíprocos de los límites de cambio de velocidad, se tiene que:

$$\frac{1}{h_i} \leq \frac{t_{i+1}}{d_{i+1}z} - \frac{t_i}{d_i z} \leq \frac{1}{g_i} \rightarrow \frac{d_i}{h_i} \leq \frac{t_{i+1}d_i}{d_{i+1}z} - \frac{t_i}{z} \leq \frac{d_i}{g_i} \rightarrow \frac{d_i}{h_i} z \leq \frac{d_i}{d_{i+1}}(t_{i+1}) - t_i \leq \frac{d_i}{g_i} z.$$

Estas expresiones corresponden a las ecuaciones (1.11), (1.12), (1.13) y (1.14) del modelo lineal 1.4.1.

**Observación 2.** *El razonamiento anterior, sirve para linealizar las restricciones de velocidad y cambio de velocidad en la dirección de entrada o inbound. De hecho, se*

tiene que:

$$\frac{\bar{d}_i}{f_i} z \leq \bar{t}_i \leq \frac{\bar{d}_i}{e_i} z \quad \text{y} \quad \frac{\bar{d}_i}{h_i} z \leq \frac{\bar{d}_i}{d_{i+1}} (\bar{t}_{i+1}) - \bar{t}_i \leq \frac{\bar{d}_i}{g_i} z.$$

Sobre los cambios máximos de velocidad, tanto para el límite superior e inferior de velocidad, estos pueden ser calculados a partir de los recíprocos de los límites de cambio de velocidad y de los límites. Para esto, note que:

$$\frac{1}{h_i} \leq \frac{1}{v_i} - \frac{1}{v_{i+1}} \leq \frac{1}{g_i} \rightarrow \frac{v_{i+1}v_i}{h_i} \leq v_{i+1} - v_i \leq \frac{v_{i+1}v_i}{g_i}.$$

De aquí que, si se requiere conocer el cambio máximo ( $\pm$ ) para el límite superior e inferior de velocidad, se debe cumplir que:

$$\frac{e_i^2}{h_i} \leq v_{i+1} - v_i \leq \frac{f_i^2}{g_i},$$

y por tanto, el cambio máximo de velocidad para el límite superior e inferior de esta, viene dado por  $\pm \frac{f_i^2}{g_i}$  y  $\pm \frac{e_i^2}{h_i}$ , respectivamente.

**Observación 3.** *El razonamiento anterior sirve para calcular el cambio máximo de velocidad, en la dirección de entrada o inbound, para los límites superior e inferior de velocidad. De hecho, estos vienen dado por  $\pm \frac{f_i^2}{g_i}$  y  $\pm \frac{e_i^2}{h_i}$ , respectivamente.*

De la geometría en la figura 1.1 se pueden deducir las siguientes ecuaciones:

$$\phi_{ij} + \bar{\phi}_{ij} + \Delta_i - \Delta_j = m_{ij} \quad (1.1)$$

donde  $m_{ij}$  es una variable de decisión entera (números de periodos).

$$\phi_{ij} + \frac{1}{2}r_j + w_j + \tau_j = \frac{1}{2}r_i + w_i + t_{ij} \quad (1.2)$$

y además,

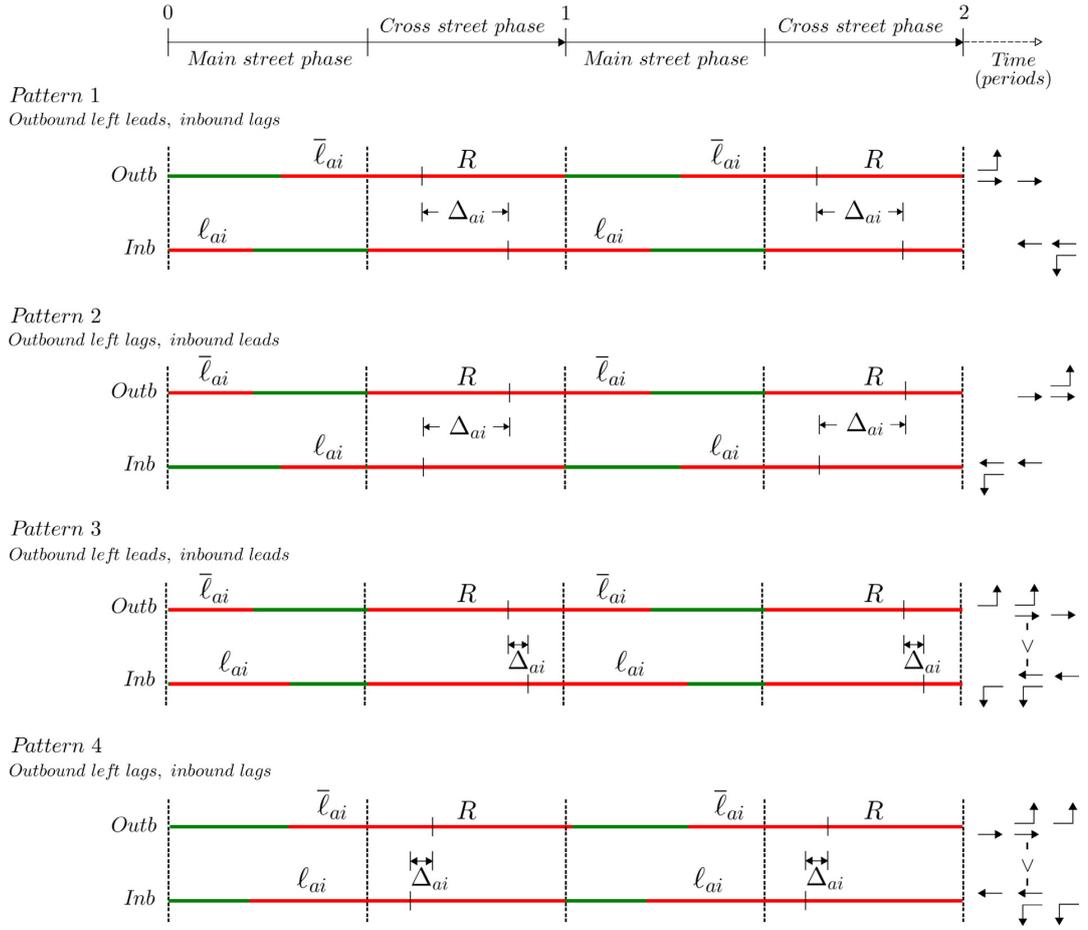
$$\bar{\phi}_{ij} + \frac{1}{2}\bar{r}_j + \bar{w}_j + \bar{\tau}_i = \frac{1}{2}\bar{r}_i + \bar{w}_i + \bar{t}_{ij} \quad (1.3)$$

Así reemplazando las ecuaciones (1.2) y (1.3) en la ecuación (1.1) se tiene la ecuación:

$$t_{ij} + \bar{t}_{ij} + \frac{1}{2}(r_i + \bar{r}_i) + (w_i + \bar{w}_i) - \frac{1}{2}(r_j + \bar{r}_j) - (w_j + \bar{w}_j) - (\tau_j + \bar{\tau}_i) + (\Delta_i - \Delta_j) = m_{ij} \quad (1.4)$$

La ecuación (1.4) se conoce como restricción de bucle arterial entre las señales  $S_i$  y  $S_j$ .

Existen restricciones que modelan las decisiones de giro a la izquierda si las luces verdes lo permiten. De hecho, la formulación MAXBAND permite decidir entre cuatro patrones posibles de giros a la izquierda que se asemejan a la figura 1.2. Aquí  $\ell_i$  y  $\bar{\ell}_i$  son parámetros que representan, para una señal  $S_i$  sobre una arteria, el tiempo (en periodos) de las fases de giro hacia la izquierda de ida y vuelta, respectivamente.



**Figura 1.2.** Patrones de fases de giro a la izquierda. Aquí  $R$  es el tiempo rojo en común. Ver (Cabezas et al., 2019).

Con base en la geometría de la figura 1.2 se puede obtener expresiones para  $\Delta_i$  en función de  $l_i$ ,  $\bar{l}_i$  y de variables binarias  $\delta_i$ ,  $\bar{\delta}_i$ . Véase la tabla 1.1.

Patrón	$\Delta_i$	$\delta_i$	$\bar{\delta}_i$
1	$-\frac{l_i + \bar{l}_i}{2}$	0	1
2	$\frac{l_i + \bar{l}_i}{2}$	1	0
3	$-\frac{l_i + \bar{l}_i}{2}$	0	0
4	$\frac{l_i + \bar{l}_i}{2}$	1	1

**Tabla 1.1.** Expresiones para  $\Delta_i$  en función de los cuatro patrones de giros hacia la izquierda.

Las expresiones de la tabla 1.1 permiten establecer que:

$$\Delta_i = \frac{1}{2}[(2\delta_i - 1)l_i - (2\bar{\delta}_i)\bar{l}_i] \quad (1.5)$$

Las decisiones sobre giros a la izquierda se incluyen en el modelo. Para este fin basta sustituir la ecuación (1.4) en la ecuación (1.5), de donde se obtiene que:

$$(w_i + \bar{w}_i) - (w_{i+1} + \bar{w}_{i+1}) + (t_i + \bar{t}_i) + (\delta_i \ell_i - \bar{\delta}_i \bar{\ell}_i) - (\delta_{i+1} \ell_{i+1} - \bar{\delta}_{i+1} \bar{\ell}_{i+1}) \\ + (r_i - r_{i+1}) - (\tau_{i+1} + \bar{\tau}_i) = m_i.$$

Véase la ecuación (1.10) del modelo lineal 1.4.1.

Para más detalles de todo lo expuesto hasta este punto del trabajo, el lector puede consultar los trabajos de Little et al. (1981), Cabezas García (2015) y Cabezas et al. (2019).

Finalmente, la propuesta general de la formulación MAXBAND para el caso arterial, introducida por Little et al. (1981), es la que sigue:

---

**LM 1.4.1** Formulación MAXBAND caso arterial

---

$$\text{Maximizar} \quad kb + \bar{k}\bar{b} \quad (1.6)$$

sujeto a :

$$\frac{1}{T_2} \leq z \leq \frac{1}{T_1}, \quad (1.7)$$

$$w_i + b \leq 1 - r_i, \quad \forall i = 1, \dots, n, \quad (1.8)$$

$$\bar{w}_i + \bar{b} \leq 1 - \bar{r}_i, \quad \forall i = 1, \dots, n, \quad (1.9)$$

$$(w_i + \bar{w}_i) - (w_{i+1} + \bar{w}_{i+1}) + (t_i + \bar{t}_i) \\ + (\delta_i \ell_i - \bar{\delta}_i \bar{\ell}_i) - (\delta_{i+1} \ell_{i+1} - \bar{\delta}_{i+1} \bar{\ell}_{i+1}) + (r_i - r_{i+1}) \\ - (\tau_{i+1} + \bar{\tau}_i) = m_i, \quad \forall i = 1, \dots, n-1, \quad (1.10)$$

$$\left( \frac{d_i}{f_i} \right) z \leq t_i \leq \left( \frac{d_i}{e_i} \right) z, \quad \forall i = 1, \dots, n-1, \quad (1.11)$$

$$\left( \frac{\bar{d}_i}{\bar{f}_i} \right) z \leq \bar{t}_i \leq \left( \frac{\bar{d}_i}{\bar{e}_i} \right) z, \quad \forall i = 1, \dots, n-1, \quad (1.12)$$

$$\left( \frac{d_i}{h_i} \right) z \leq \left( \frac{d_i}{d_{i+1}} \right) t_{i+1} - t_i \leq \left( \frac{d_i}{g_i} \right) z, \quad \forall i = 1, \dots, n-2, \quad (1.13)$$

$$\left( \frac{\bar{d}_i}{\bar{h}_i} \right) z \leq \left( \frac{\bar{d}_i}{\bar{d}_{i+1}} \right) \bar{t}_{i+1} - \bar{t}_i \leq \left( \frac{\bar{d}_i}{\bar{g}_i} \right) z, \quad \forall i = 1, \dots, n-2, \quad (1.14)$$

$$m_i \in \mathbb{Z}, \quad \forall i = 1, \dots, n-1, \quad (1.15)$$

$$\delta_i, \bar{\delta}_i \in \{0, 1\}, \quad \forall i = 1, \dots, n-1, \quad (1.16)$$

$$b, \bar{b}, t_i, \bar{t}_i, w_i, \bar{w}_i, z \geq 0, \quad \forall i = 1, \dots, n-1. \quad (1.17)$$


---

### 1.4.3. Ciclos y base de ciclos

Sea  $G = (V, E)$  un grafo no dirigido donde  $V$ , respectivamente  $E$ , representa el conjunto de vértices, respectivamente aristas, digamos  $\{v_1, v_2, \dots, v_n\}$ , respectivamente  $\{e_1, e_2, \dots, e_m\}$ , de  $G$ . Sobre  $V$ , respectivamente sobre  $E$ , es posible definir un  $\mathbf{F}_2$ -espacio vectorial donde  $\mathbf{F}_2 = \{0, 1\}$ . En efecto, basta considerar las operaciones  $+$  :  $\mathcal{V}(G) \times \mathcal{V}(G) \rightarrow \mathcal{V}(G)$  y  $\cdot$  :  $\mathbf{F}_2 \times \mathcal{V}(G) \rightarrow \mathcal{V}(G)$ , respectivamente

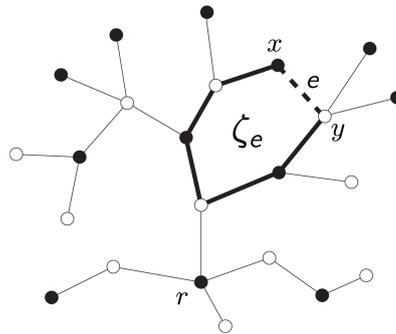
$+$  :  $\mathcal{E}(G) \times \mathcal{E}(G) \rightarrow \mathcal{E}(G)$  y  $\cdot$  :  $\mathbf{F}_2 \times \mathcal{E}(G) \rightarrow \mathcal{E}(G)$ , donde  $\mathcal{V}(G)$ , respectivamente  $\mathcal{E}(G)$  es el conjunto potencia de  $E$ , respectivamente  $V$ , y  $+$ , respectivamente  $\cdot$ , es la diferencia simétrica de conjuntos, respectivamente es tal que  $1 \cdot x = x$  y  $0 \cdot x = \emptyset$  para todo  $x \in \mathcal{V}(G)$  y  $x \in \mathcal{E}(G)$ . Nótese que  $\{\{v_1\}, \{v_2\}, \dots, \{v_n\}\}$ , respectivamente  $\{\{e_1\}, \{e_2\}, \dots, \{e_m\}\}$ , es una base-conocida como base canónica-para  $\mathcal{V}(G)$ , respectivamente  $\mathcal{E}(G)$ . Para mayor información, se referencia al lector al trabajo de Diestel (2005).

Un subespacio importante y de interés sobre  $G$  es el espacio de ciclos  $\mathcal{C}(G)$ , el cual es generado por todos los ciclos de  $G$ . Aún es de más interés (en particular para el presente trabajo) encontrar bases para  $\mathcal{C}(G)$ .

En la literatura se pueden encontrar algunas clases de bases para  $\mathcal{C}(G)$ , véase, por ejemplo, (Liebchen et al., 2007) donde, entre otras cosas, se presenta un resumen clasificadorio de varios tipos de bases para  $\mathcal{C}(G)$  (hasta ese entonces) en función de algunos tipos de grafos, entre ellos, no dirigidos.

Algunas clases de base de ciclos que podrían utilizarse en el presente trabajo son las siguientes:

- *Base de ciclo estrictamente fundamental*: Un conjunto  $B$  de ciclos de un grafo no dirigido  $G = (V, E)$  se dice que es una *base de ciclo estrictamente fundamental*, si existe algún árbol generador  $T \subset E$  tal que  $B = \{\zeta_e \mid e \in E \setminus T\}$  donde  $\zeta_e$  denota el único ciclo de  $T \cup \{e\}$ . (Liebchen et al., 2007, Definición 15)

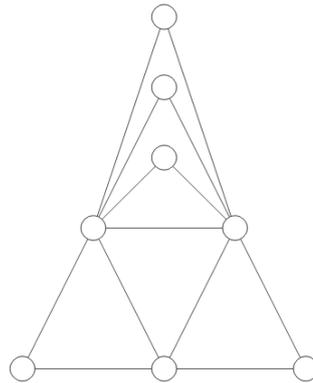


**Figura 1.3.** El ciclo  $\zeta_e$  en  $T \cup e$ . (Diestel, 2005).

- *Base de ciclo débilmente fundamental*: Un conjunto  $B = \zeta_1, \dots, \zeta_v$  de ciclos de un grafo no dirigido  $G = (V, E)$  es una *base de ciclo débilmente fundamental* si existe alguna permutación  $\sigma$  tal que

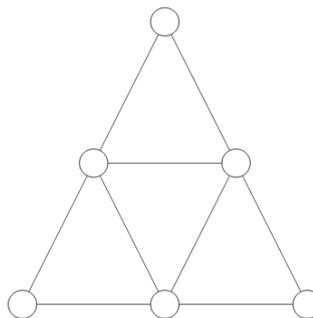
$$\zeta_{\sigma(i)} \setminus (\zeta_{\sigma(1)} \cup \dots \cup \zeta_{\sigma(i-1)}) \neq \emptyset \quad \forall i = 2, \dots, v.$$

En otras palabras, obsérvese que  $B$  sería una base de ciclo débilmente fundamental si no existe arista alguna  $e \in E$  de manera que esta esté contenida en al menos tres ciclos de  $B$ . (Liebchen et al., 2007, Definición 9, Lema 10). Véase la figura 1.4.



**Figura 1.4.** Base de ciclo débilmente fundamental formada por los seis triángulos visibles.

- *2-Base*: Un conjunto  $B$  de ciclos de un grafo no dirigido  $G = (V, E)$  se dice que es una *2-Base* si es una base de ciclo y además cada arista de  $e \in E$  está contenida a lo más en dos ciclos de  $B$ . (Liebchen et al., 2007, Definición 11). Véase la figura 1.5.



**Figura 1.5.** Base de ciclo formada con los cuatro triángulos visibles.

Los siguientes resultados describen la naturaleza de la relación subyacente entre las tres bases de ciclos mencionadas:

**Teorema 1.** (Liebchen et al., 2007, Corolario 18) *Toda base de ciclo estrictamente fundamental es débilmente fundamental.*

**Teorema 2.** (Liebchen et al., 2007, Lema 18) *Toda 2-base es una base de ciclo débilmente fundamental.*

En muchas aplicaciones, cada arista  $e \in E$  de un grafo  $G(V, E)$  tiene un valor numérico  $w(e)$  asociado, llamado peso. Por ejemplo, en una red de transporte, podría representar las distancias entre nodos, el costo de llevar una carga entre nodos, etc.

Sea  $\zeta$  un ciclo del algún grafo. El peso de  $\zeta$  se define como:

$$w(\zeta) = \sum_{e \in \zeta} w(e).$$

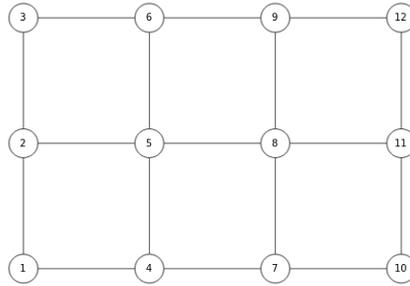
Más aún, el peso de una base de ciclos  $B$ , se define como:

$$w(B) = \sum_{\zeta \in B} \sum_{e \in \zeta} w(e)$$

El peso de una base de ciclo será de interés en el estudio del impacto de una base de ciclos sobre la implementación computacional de la formulación MAXBAND.

#### 1.4.4. Formulación MAXBAND sobre una red

La generalización natural de la formulación MAXBAND caso arterial es sobre una red. Por ejemplo los *grid graphs* representan muchos de los problemas de redes en la vida real, ver figura 1.6. La restricción del bucle arterial, la ecuación (1.4), puede verse como un ciclo para dos nodos porque representa el movimiento de ida y vuelta de una señal. Al extender esta idea sobre ciclos más grandes, es claro que la suma de todas las compensaciones en el ciclo también debe ser un número entero.



**Figura 1.6.** Grid graph  $G_{3 \times 4}(E, V)$ .

Sea  $A$  el conjunto de arterias sobre una red. Cada arteria  $a \in A$  tiene señales  $S_{aj}$  donde  $j$  es el índice de señal  $j$ -ésima sobre la arteria. Para obtener una restricción, dado cualquier ciclo  $\zeta$  sobre una red, se consideran los siguientes dos conjuntos:

1.  $A_{\zeta}^F(A_{\zeta}^B)$ : Conjunto de todos los segmentos de arterias delanteras (traseras) con bordes de arista  $(i, j)$  en el sentido de las agujas del reloj del ciclo  $\zeta$ .
2.  $J_{\zeta}$ : Todos los conjuntos de la forma  $(b, j, i, c, k)$  en  $\zeta$  donde  $i$  es la unión entre las arterias  $b$  y  $c$  en las señales  $S_{bj}$  y  $S_{ck}$ .

Así, la restricción de bucle de red (restricción de ciclo) puede ser escrita como:

$$\sum_{(i,j):a \in A_{\zeta}^F} \phi_{ij}^a - \sum_{(i,j):a \in A_{\zeta}^B} \phi_{ij}^a + \sum_{(b,j,i,c,k) \in J_{\zeta}} \Psi_{S_{bj}, S_{ck}}^i = C_{\zeta}$$

donde  $C_{\zeta}$  es una variable de decisión entera y  $\Psi_{S_{bj}, S_{ck}}^i$  es la variable de decisión expresada como compensación intranodo que representa el tiempo entre los centros consecutivos de rojos para las señales  $S_{bj}$  y  $S_{ck}$ .

Así, el modelo general sobre una red, incluyendo las restricciones de ciclo, es:

**LM 1.4.2** Formulación MAXBAND sobre una red

$$\text{Maximizar} \quad \sum_{a \in A} (k_a b_a + \bar{k}_a \bar{b}_a) \quad (1.18)$$

sujeto a :

$$\frac{1}{T_2} \leq z \leq \frac{1}{T_1}, \quad (1.19)$$

$$w_{ai} + b_a \leq 1 - r_{ai}, \quad \forall a \in A, \forall i = 1, \dots, n_a, \quad (1.20)$$

$$\bar{w}_{ai} + \bar{b}_a \leq 1 - \bar{r}_{ai}, \quad \forall a \in A, \forall i = 1, \dots, n_a, \quad (1.21)$$

$$\begin{aligned} & (w_{ai} + \bar{w}_{ai}) - (w_{a,i+1} + \bar{w}_{a,i+1}) + (t_i^a + \bar{t}_i^a) \\ & + (\delta_{ai} \ell_{ai} - \bar{\delta}_{ai} \bar{\ell}_{ai}) - (\delta_{a,i+1} \ell_{a,i+1} - \bar{\delta}_{a,i+1} \bar{\ell}_{a,i+1}) + (r_{ai} - r_{a,i+1}) \\ & - (\tau_{a,i+1} + \bar{\tau}_{ai}) = m_i^a, \quad \forall a \in A, \forall i = 1, \dots, n_a - 1, \quad (1.22) \end{aligned}$$

$$\left( \frac{d_i^a}{f_i^a} \right) z \leq t_i^a \leq \left( \frac{d_i^a}{e_i^a} \right) z, \quad \forall a \in A, \forall i = 1, \dots, n_a - 1, \quad (1.23)$$

$$\left( \frac{\bar{d}_i^a}{\bar{f}_i^a} \right) z \leq \bar{t}_i^a \leq \left( \frac{\bar{d}_i^a}{\bar{e}_i^a} \right) z, \quad \forall a \in A, \forall i = 1, \dots, n_a - 1, \quad (1.24)$$

$$\left( \frac{d_i^a}{h_i^a} \right) z \leq \left( \frac{d_{i+1}^a}{d_{i+1}^a} \right) t_{i+1}^a - t_i^a \leq \left( \frac{d_i^a}{g_i^a} \right) z, \quad \forall a \in A, \forall i = 1, \dots, n_a - 2, \quad (1.25)$$

$$\left( \frac{\bar{d}_i^a}{\bar{h}_i^a} \right) z \leq \left( \frac{\bar{d}_i^a}{\bar{d}_{i+1}^a} \right) \bar{t}_{i+1}^a - \bar{t}_i^a \leq \left( \frac{\bar{d}_i^a}{\bar{g}_i^a} \right) z, \quad \forall a \in A, \forall i = 1, \dots, n_a - 2, \quad (1.26)$$

$$\sum_{(i,j): a \in A_\zeta^F} \phi_{ij}^a - \sum_{(i,j): a \in A_\zeta^B} \phi_{ij}^a + \sum_{(b,j,i,c,k) \in J_\zeta} \Psi_{S_{bj}, S_{ck}}^i = C_\zeta, \quad \forall \zeta \in \mathcal{B}_\zeta, \quad (1.27)$$

$$C_\zeta \in \mathbb{Z}, \quad \forall \zeta \in \mathcal{B}_\zeta, \quad (1.28)$$

$$m_i^a \in \mathbb{Z}, \quad \forall a \in A, \forall i = 1, \dots, n_a - 1, \quad (1.29)$$

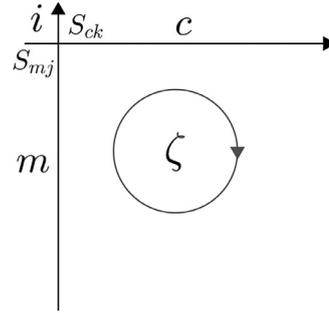
$$\delta_{ai}, \bar{\delta}_{ai} \in \{0, 1\}, \quad \forall a \in A, \forall i = 1, \dots, n_a - 1, \quad (1.30)$$

$$b_a, \bar{b}_a, t_i^a, \bar{t}_i^a, w_{ai}, \bar{w}_{ai}, z \geq 0, \quad \forall a \in A, \forall i = 1, \dots, n_a - 1. \quad (1.31)$$

En general, un grid graph  $G_{k \times k}$  requiere  $(k-1)^2$  restricciones de bucle de red y, en consecuencia, se deben calcular varios intranodos para cada una de estas ecuaciones. Los valores de las compensaciones intranodo  $\Psi_{S_{bj}, S_{ck}}^i$  dependen de las posiciones de tiempo rojas de las señales  $S_{bj}$  y  $S_{ck}$  en una calle principal y una calle transversal, respectivamente. En el trabajo de Cabezas Cabezas et al. (2019) se encuentra una forma elegante, en función de los parámetros  $\ell$  y variables de decisión  $\delta$ , de calcular los valores de las compensaciones intranodo.

**Teorema 3.** (Cabezas et al., 2019, Teorema 1) Considere los patrones de las fases de giro a la izquierda que se muestran en la figura 1.2. Sea  $\Psi_{mc}^{p_m p_c}$  el valor de  $\Psi_{S_{mj}, S_{ck}}^i$  de las arterias  $m$  y  $c$  que se encuentran en la unión  $i$  para señales  $S_{mj}$  y  $S_{ck}$  con patrones de fases de giro a la izquierda  $p_m$  y  $p_c$ , respectivamente (Ver figura 1.7). Entonces, para todos los valores posibles de  $p_m$  y  $p_c$ , se tiene que:

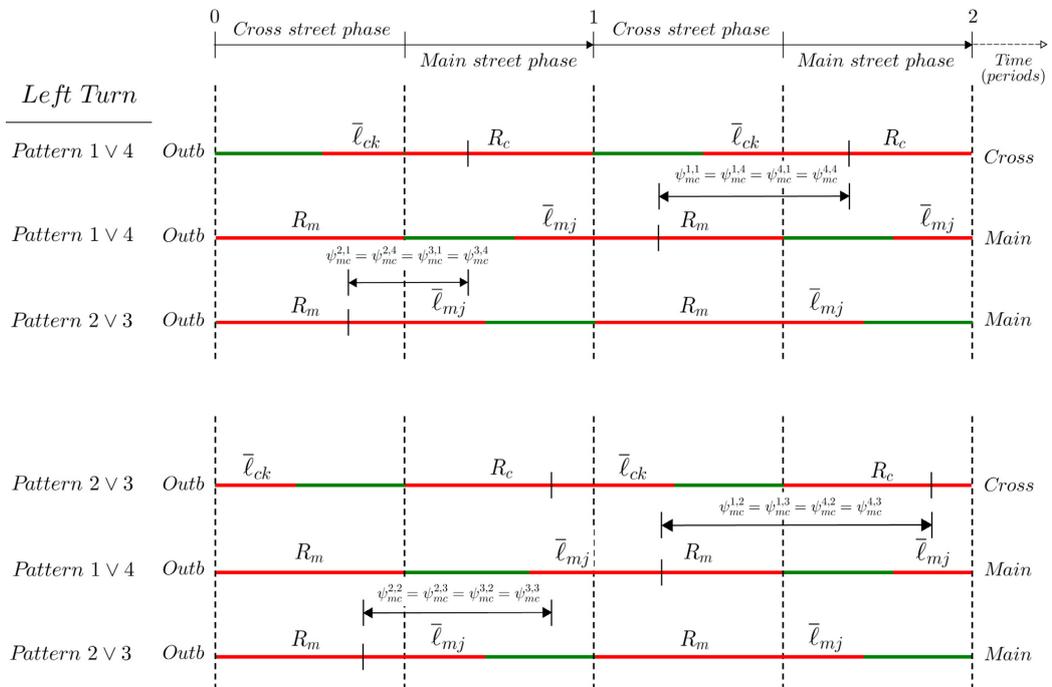
$$\Psi_{S_{mj}, S_{ck}}^i = \frac{1}{2} - \frac{1}{2} [(2\bar{\delta}_{mk} - 1)\bar{\ell}_{mk} - (2\delta_{mk} - 1)]$$



**Figura 1.7.** Unión  $i$  de las arterias principal ( $m$ ) y transversal ( $c$ ). Véase (Cabezas et al., 2019).

**Observación 4.** Es bien conocido que el número de restricciones de ciclo en el modelo depende de la cantidad de aristas y nodos de la red. Si este número (de restricciones) es muy grande (que lo puede ser) el problema será muy difícil de resolver. Como se mencionó en la sección 1.4.3, el conjunto de todos los ciclos  $\zeta$  en cualquier grafo puede generarse mediante una base  $B_\zeta^1$  (base de ciclos) y más aún, esta base tiene cardinalidad  $m - n + 1$ , donde  $m$  representa el número de aristas y  $n$  el número de nodos en el grafo no dirigido subyacente relacionado con el grafo dirigido que representa la red original. Por lo tanto, es sensato considerar que se debe encontrar una base de ciclo para la formulación MAXBAND sobre una red.

Finalmente, se puede notar en la figura 1.8, que muestra la idea general implícita de la geometría para  $\Psi_{mc}^{PmPc}$ .



**Figura 1.8.** Geometría de  $\Psi_{mc}^{PmPc}$ . Ver (Cabezas et al., 2019).

<sup>1</sup>Los ciclos considerados en la restricción 1.27 del modelo 2.2.1, son precisamente los que pertenecen a la base de ciclos  $B_\zeta$ .

### 1.4.5. Una breve introducción a la teoría de Webster

En los casos generales de la formulación MAXBAND se requiere información para la división del tiempo de las luces verdes que permitan giros a la izquierda en las intersecciones entre arterias, véase los modelos 1.4.1 y 1.4.2. Existen cuatro posibles patrones de fases de giro a la izquierda, véase la figura 1.2. Note que la formulación MAXBAND modelan las decisiones de giro a la izquierda si las luces verdes lo permiten.

---

#### Algorithm 1 Algoritmo de Webster

---

##### INPUT:

- $N$ : Conjunto de fases del ciclo semafórico.
- $G_i$ : Conjunto de grupos de carriles de la fase  $i \in N$ .
- $v_{ik}$ : Flujo vehicular del grupo de carriles  $k \in G_i$  en el grupo de carriles  $i$
- $s_{ik}$ : Flujo de saturación del grupo de carriles  $k \in G_i$  en el grupo de carriles  $i$
- $t_i$ : Tiempo perdido por fase  $i \in N$ .
- $A_i$ : Intervalo de tiempo del amarillo en la fase  $i \in N$ .
- $AR_i$ : Intervalo de tiempo del rojo en la fase  $i \in N$ .

**OUTPUT:** Intervalo en verde,  $G_i$ , en cada fase del ciclo semafórico.

```

1: for  $i = 1$  to  $|N|$  do
2:   for  $k = 1$  to  $|G_i|$  do
3:     Calcule  $\frac{v_{ik}}{s_{ik}}$ 
4:   end for
5:    $Y_i = \max \left\{ \frac{v_{i1}}{s_{i1}}, \frac{v_{i2}}{s_{i2}}, \dots, \frac{v_{i|G_i|}}{s_{i|G_i|}} \right\}$ 
6: end for
7:  $Y = \sum_{i \in N} Y_i$ 
8:  $L = \sum_{i \in N} t_i$ 
9:  $C = \frac{1.5L+5}{1-Y}$ 
10:  $g_T = C - L$ 
11: for  $i = 1$  to  $|N|$  do
12:    $g_i = \frac{Y_i}{Y} g_T$ 
13:    $G_i = g_i - A_i - AR_i + t_i$ 
14: end for

```

---

Una opción en la formulación MAXBAND es que el usuario proporcione información para la división del tiempo de las luces verdes a su criterio. Sin embargo, es posible determinar una división óptima de estos tiempos a partir de información sobre

el volumen y capacidad del tráfico de las arterias involucradas en una intersección. La teoría principal utilizada para este propósito es la teoría de Webster (1958), la cual brinda un algoritmo para minimizar el tiempo de ciclo que da el menor retraso a todos los vehículos que utilizan una intersección aislada<sup>2</sup>. Ver Algoritmo 1.

Para aplicar el algoritmo de Webster en la formulación MAXBAND, se tiene que  $N$  es el conjunto de las cuatro fases posibles descritas en la figura 1.2, el conjunto  $G_i$  consta de dos carriles (entrada y salida) y  $A_i$  se considera como parte del rojo ( $AR_i$ ).

En general,  $v_{ik}$  y  $s_{ik}$  se encuentran en unidades de vehículos por hora. La diferencia entre  $s_{ik}$  y  $v_{ik}$  es que representan el flujo vehicular máximo y real del grupo de carriles, respectivamente. Tanto  $t_i$ ,  $A_i$  y  $AR_i$  se encuentran en segundos.

Ahora bien,  $g_T$  representa el tiempo efectivo de luz verde para todo el sistema de fases,  $g_i$  es el tiempo efectivo de luz verde por fase y  $G_i$  es la duración total de luz verde por fase.

Lo que pretende el algoritmo de Webster es minimizar el tiempo de demora en las intersecciones, dividiendo el tiempo de ciclo disponible entre el flujos de tráfico de una intersección en relación proporcional a sus volúmenes dividido por sus capacidades.

---

<sup>2</sup>Una intersección es aislada si dentro de una vecindad de la intersección (500m, 1km, etc. a la redonda) no hay más semáforos

# Capítulo 2

## Metodología

En las primeras formulaciones MAXBAND, Morgan et al. (1964) implementaron dos algoritmos sistemáticos (no lineales) para obtener una solución proveniente de la formulación. Esos dos algoritmos son el **SEB** (Synchronization for Equal Bandwidths) y **SUB** (Synchronization for Unequal Bandwidths). Posteriormente, Little et al. (1981) presentaron una formulación del modelo MAXBAND como un problema de programación lineal mixta. La formulación lineal del modelo MAXBAND resultó ser más eficiente que los algoritmos **SUB** y **SEB**.

Ahora bien, ya para 1981 se tenía una formulación del modelo MAXBAND como un problema de programación lineal mixto, pero aún así no consentía el caso general de una red, de hecho, se llegó hasta la formulación sobre una red de tres arterias con diecisiete señales, véase (Little et al., 1981). Fue en años posteriores que la formulación MAXBAND como problema de programación lineal mixta se generaliza sobre cualquier red. Cabezas et al. (2019) resumen este trabajo, y entre otras aportaciones, proponen una heurística híbrida para dicha formulación.

En el presente manuscrito se implementó la formulación lineal MAXBAND descrita por Cabezas et al. (2019) en *Python 3.8.5* Van Rossum et al. (2009) utilizando el paquete para optimización *Pyomo* Bynum et al. (2021) y el solver *CPLEX* Cplex (2009), porque a diferencia de otros programas, modelos y formulaciones:

- El lenguaje de programación Python es un lenguaje muy usado (superando a C y Java), comercial, de licencia libre y está en constante actualización. Pyomo, por otra parte, es software de Python especializado en modelos de optimización que puede usar solvers rígidos. Todo esto permite que la implementación sea de fácil aplicación y pueda ser constantemente mejorada (actualizada) por expertos de varias empresas alrededor del mundo.
- El problema de optimización a resolver es un problema lineal. Los problemas lineales son los “preferidos” de los computadores ya que son más fáciles y rápidos de resolver. A pesar de que el problema de sincronización de semáforos en una red es un problema NP- duro, esta implementación supone una ventaja computacional.
- Se adapta como un modelo muy acorde a la realidad del problema de sincronización de semáforos. La aplicación de base de ciclos en este tipo de modelos

es una idea innovadora y bien justificada para este tipo de problemas.

Sobre la metodología general del presente trabajo, y con el fin de conseguir nuestro objetivo:

1. Se implementó (en Pyomo-Python) la formulación MAXBAND caso arterial, suponiendo que las luces rojas se superponen, no hay tiempo de liberación para líneas de espera (pelotones de carros) y velocidades preasignadas entre semáforos. **Caso arterial 1.**
2. Se implementó (en Pyomo-Python) la formulación MAXBAND como en el caso anterior, salvo que se consideró velocidades variables (pero acotadas) entre semáforos. **Caso arterial 2.**
3. Se implementó (en Pyomo-Python) la formulación MAXBAND como en el caso anterior, salvo que se consideró giros hacia la izquierda<sup>1</sup>. **Caso arterial 3.**
4. Una vez implementado el caso general sobre una arteria, se procedió a implementar el caso sobre una red. Para esto se introducen las restricciones de bucle. **Caso general sobre una red.**
5. Es preciso comentar que en cada caso se realizaron simulaciones con el fin de verificar el correcto funcionamiento de las implementaciones. Finalmente, se simuló el caso general sobre una red con datos reales de ciertas calles de la ciudad de Guayaquil, Ecuador.

## 2.1. Sobre la sincronización de semáforos: Formulación MAXBAND

Para calcular la sincronización de semáforos que maximiza el bandwidth, a partir de la output de la formulación MAXBAND, se introduce la siguiente variable:

- $\theta(h, i)$ : Fase relativa, o compensación, de  $S_h$  y  $S_i$ , medida como el tiempo desde el centro de un rojo de  $S_h$  hasta el próximo centro de un rojo de  $S_i$  en ciclos.

Formalmente, el concepto de sincronización de semáforos (en el contexto de este trabajo) es el que sigue:

**Definición 2.** (Morgan et al., 1964, Sincronización de semáforos) Sean  $S_1, S_2, \dots, S_n$  señales (semáforos) sobre una arteria. Se conoce como sincronización al conjunto  $\{\theta(h, i) \mid i = 1, 2, 3, \dots, n\}$  para  $h \in \{1, 2, 3, \dots, n\}$ .

---

<sup>1</sup>No se consideran giros a la derecha ya que, por lo general, se permiten giros a la derecha aún en luz roja, esto siempre y cuando se gire con precaución.

En la literatura, por convención,  $0 \leq \theta(h, i) \leq 1$ . Más aún, por la geometría de la figura 1.1,

$$\theta(h, i) = \text{man}(\phi(h, i))$$

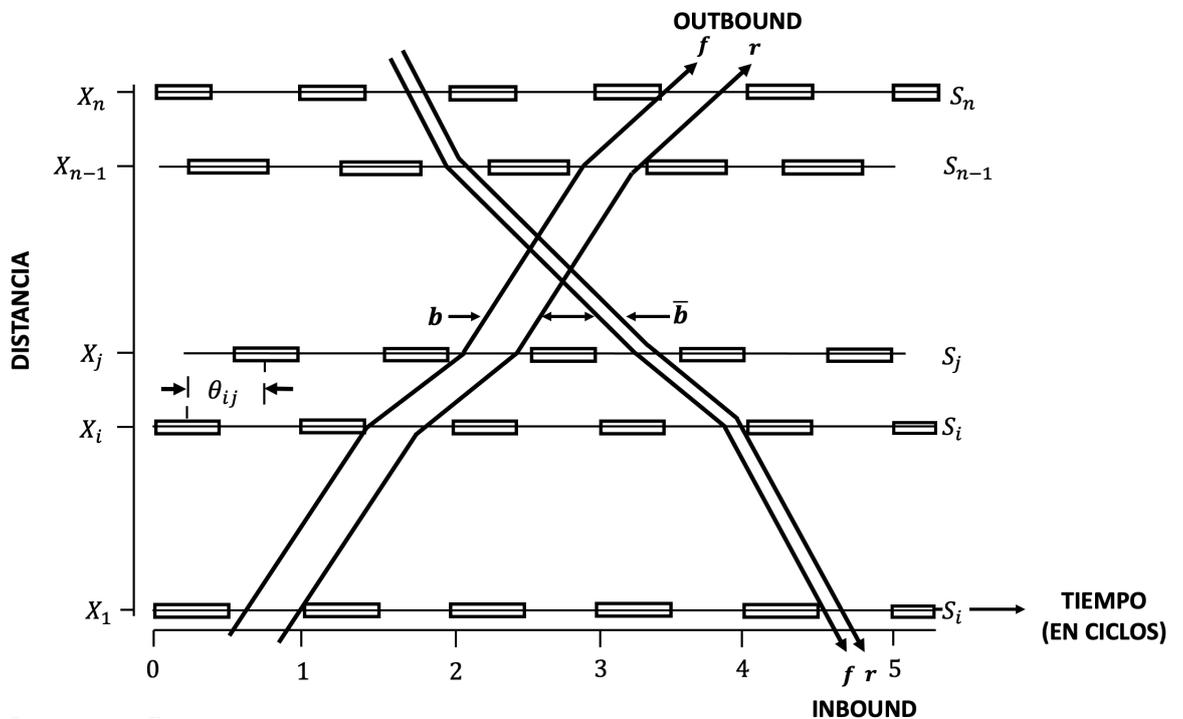
donde  $\text{man}(x)$  representa la *mantisa* de un número (real)  $x$ . La mantisa de  $x$ , se obtiene al eliminar la parte entera de  $x$  y, si el resultado es negativo, se suma uno. Equivalentemente,  $\text{man}(x) = \text{mod}(x, 1)$ , donde  $\text{mod}(x, n) = x - n \cdot \lfloor x/n \rfloor$ , tal como se define en (Knuth et al., 1973).

Para  $h \in \{1, 2, 3, \dots, n\}$  fijo, digamos  $h = 1$ , se tiene que:

$$\theta(1, i) = \text{man} \left( w_1 - w_i + \sum_{k=1}^{i-1} t_k + \frac{1}{2}(r_1 - r_i) \right) \quad (2.1)$$

En principio, la ecuación (2.1) se obtiene suponiendo que los  $\Delta$ 's y  $\tau$ 's son iguales a cero. Sin embargo, en los casos más generales, estos valores,  $\theta(h, i)$ , se obtienen de forma similar.

Finalmente, la sincronización nos indica *cómo ajustar* óptimamente las señales de los distintos semáforos a lo largo de una arteria en función del tiempo (ciclos), para que el bandwidth óptimo, obtenido de la formulación MAXBAND, pueda ser aplicado, véase la figura 2.1. Para más detalles, el lector puede revisar los códigos anexos y ver cómo se aplica este concepto en la obtención de las gráficas de la sincronización de semáforos.



**Figura 2.1.** Diagrama espacio-tiempo donde se ilustra la sincronización de semáforos óptima en función de  $\theta_{ij}$  y los bandwidth's de entrada y salida. Ver (Little et al., 1981).

## 2.2. Modelo lineal: caso arterial 1

Para la implementación de este modelo, al igual que el SEB, se consideró los bandwidths (de entrada y salida iguales), es decir  $b = \bar{b}$ . Más aún, las restricciones se redujeron comparado con el modelo MAXBAND expuesto en 1.4.1, pues en este caso las velocidades se consideraron constantes, los  $\Delta$ 's y  $\tau$ 's con valor cero, quedando la siguiente formulación:

---

### LM 2.2.1 MAXBAND: caso arterial 1

---

$$\text{Maximizar} \quad b + \bar{b} \quad (2.2)$$

sujeto a :

$$\bar{b} = b, \quad (2.3)$$

$$w_i + b \leq 1 - r_i, \quad \forall i = 1, \dots, n, \quad (2.4)$$

$$\bar{w}_i + \bar{b} \leq 1 - \bar{r}_i, \quad \forall i = 1, \dots, n, \quad (2.5)$$

$$(w_i + \bar{w}_i) - (w_{i+1} + \bar{w}_{i+1}) + (t_i + \bar{t}_i) = m_i - (r_i - r_{i+1}), \quad \forall i = 1, \dots, n-1, \quad (2.6)$$

$$m_i \in \mathbb{Z}, \quad \forall i = 1, \dots, n-1, \quad (2.7)$$

$$b, \bar{b}, w_i, \bar{w}_i \geq 0, \quad \forall i = 1, \dots, n-1. \quad (2.8)$$


---

Para la simulación de este caso, se consideró diez señales (semáforos) a lo largo de una arteria con los siguientes parámetros: las señales están ubicadas en 0, 550, 1250, 2350, 3050, 3850, 4500, 4900, 5600, 6050 metros, respectivamente. Los tiempos de los rojos (entrada y salida) son: 0,47, 0,40, 0,40, 0,47, 0,48, 0,42, 0,40, 0,40, 0,40, 0,42 ciclos. Se consideraron velocidades (de entrada y salida) preasignadas, entre señales. Estas velocidades son: 20, 50, 70, 50, 50, 50, 50, 50, 50, 50 metros/segundos para la dirección de salida y 50, 70, 35, 50, 50, 50, 50, 50, 30, metros/segundos en la dirección de entrada. Considerando una longitud de periodo de 65 segundos para la luz verde y roja, los tiempos de recorrido entre señales preasignados son: 0,17, 0,15, 0,48, 0,22, 0,25, 0,20, 0,12, 0,36, 0,13 ciclos para la dirección de entrada y 0,42, 0,22, 0,24, 0,22, 0,25, 0,2, 0,12, 0,22, 0,14 ciclos para la dirección de salida.

## 2.3. Modelo lineal: caso arterial 2

Una vez programado el modelo 2.2.1, se añadió la restricción de velocidades variables entre semáforos, limitadas por las cotas superior e inferior de velocidad y cambio de velocidad. También se utilizó el supuesto que  $t_i = \bar{t}_i$  y  $\bar{b} = b$ , obteniendo lo siguiente:

**LM 2.3.1** MAXBAND: caso arterial 2 (velocidad variable)

$$\text{Maximizar} \quad \bar{b} + b \quad (2.9)$$

sujeto a :

$$b = \bar{b} \quad (2.10)$$

$$\frac{1}{T_2} \leq z \leq \frac{1}{T_1}, \quad (2.11)$$

$$w_i + b \leq 1 - r_i, \quad \forall i = 1, \dots, n, \quad (2.12)$$

$$w_i - w_{i+1} + t_i = \frac{1}{2}m_i - \frac{1}{2}(r_i - r_{i+1}), \quad \forall i = 1, \dots, n-1, \quad (2.13)$$

$$\left(\frac{d_i}{f_i}\right) z \leq t_i \leq \left(\frac{d_i}{e_i}\right) z, \quad \forall i = 1, \dots, n-1, \quad (2.14)$$

$$\left(\frac{d_i}{h_i}\right) z \leq \left(\frac{d_i}{d_{i+1}}\right) t_{i+1} - t_i \leq \left(\frac{d_i}{g_i}\right) z, \quad \forall i = 1, \dots, n-2, \quad (2.15)$$

$$m_i \in \mathbb{Z}, \quad \forall i = 1, \dots, n-1, \quad (2.16)$$

$$b, \bar{b}, w_i, z \geq 0, \quad \forall i = 1, \dots, n-1. \quad (2.17)$$

Para la simulación de este caso se utilizó los parámetros del ejemplo arterial en (Little, 1966). Las señales están ubicadas en 0, 168, 381, 716, 929, 1173, 1371, 1493, 1706, 1843 metros, respectivamente. Los tiempos de los rojos son: 0,47, 0,40, 0,40, 0,47, 0,48, 0,42, 0,40, 0,40, 0,40, 0,42 ciclos. Los límites de los períodos son  $T_1 = 55$  segundos y  $T_2 = 75$  segundos. Los límites de velocidad inferior y superior son  $e_i = \bar{e}_i = 13,4$  metros/ segundos y  $f_i = \bar{f}_i = 17,9$  metros/segundos. Los límites del recíproco de cambio de velocidad son:  $\frac{1}{g_i} = \frac{1}{\bar{g}_i} = -\frac{1}{h_i} = -\frac{1}{\bar{h}_i} = 0,0121$  (metros/segundos)<sup>-1</sup> para cada  $i$ . Esto corresponde a que el cambio máximo de velocidad de  $\pm 2,2$  (metros/segundos) en el límite de velocidad inferior y un cambio máximo de  $\pm 3,9$  (metros/segundos) en el límite de velocidad superior.

## 2.4. Modelo lineal: caso arterial 3

Como variante en este caso se añadieron los giros a la izquierda, en el cual se modifica ligeramente la expresión (2,12) del caso 2.3.1. Se sabe que  $\Delta_i$  es el tiempo desde el centro de  $\bar{r}_i$  al centro de  $r_i$ , al añadir los giros a la izquierda la expresión puede verse como  $\Delta_i = \frac{1}{2}[(2\delta_i - 1)\ell_i - (2\bar{\delta}_i - 1)\bar{\ell}_i]$ , donde  $\delta_i, \bar{\delta}_i \in \{0, 1\}$  son variables binarias. Como se toma en cuenta los giros a la izquierda el  $\Delta_i$  será distinto de cero. Se sigue el supuesto que  $t_i = \bar{t}_i$  y  $r_i = \bar{r}_i$ . Así se tiene:

---

**LM 2.4.1** MAXBAND: caso arterial 3 (velocidad variable)

---

Maximizar  $\bar{b} + b$  (2.18)

sujeto a :

$$b = \bar{b}, \quad (2.19)$$

$$\frac{1}{T_2} \leq z \leq \frac{1}{T_1}, \quad (2.20)$$

$$w_i + b \leq 1 - r_i, \quad \forall i = 1, \dots, n, \quad (2.21)$$

$$2(w_i - w_{i+1}) + 2t_i + (\delta_i - \bar{\delta}_i)\ell_i - (\delta_{i+1} - \bar{\delta}_{i+1})\ell_{i+1} = m_i - (r_i - r_{i+1}), \quad \forall i = 1, \dots, n - 1, \quad (2.22)$$

$$\left(\frac{d_i}{f_i}\right) z \leq t_i \leq \left(\frac{d_i}{e_i}\right) z, \quad \forall i = 1, \dots, n - 1, \quad (2.23)$$

$$\left(\frac{d_i}{h_i}\right) z \leq \left(\frac{d_i}{d_{i+1}}\right) t_{i+1} - t_i \leq \left(\frac{d_i}{g_i}\right) z, \quad \forall i = 1, \dots, n - 2, \quad (2.24)$$

$$m_i \in \mathbb{Z}, \quad \forall i = 1, \dots, n - 1, \quad (2.25)$$

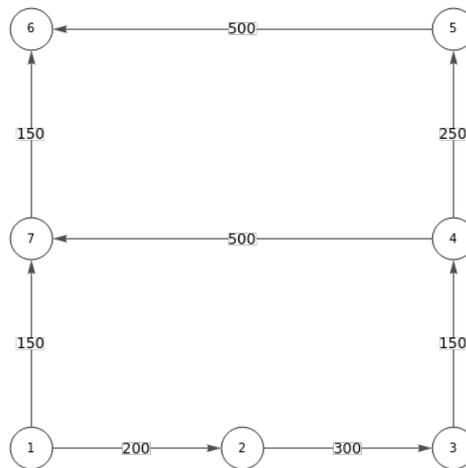
$$\delta_i, \bar{\delta}_i \in \{0, 1\} \quad \forall i = 1, \dots, n - 1, \quad (2.26)$$

$$b, \bar{b}, w_i, z \geq 0, \quad \forall i = 1, \dots, n - 1. \quad (2.27)$$


---

La aplicación del caso arterial 3, utiliza los mismos datos del ejemplo del caso arterial 2 obtenido del paper de Little (1966). Sin embargo, se añade que los parámetros que representan el tiempo (en períodos) de las fases de giro a la izquierda (de entrada y salida) se toman como un porcentaje del rojo general, particularmente, se consideró que  $\ell_j = \bar{\ell}_j = 0,3 \cdot r_j = 0,3 \cdot \bar{r}_j$ .

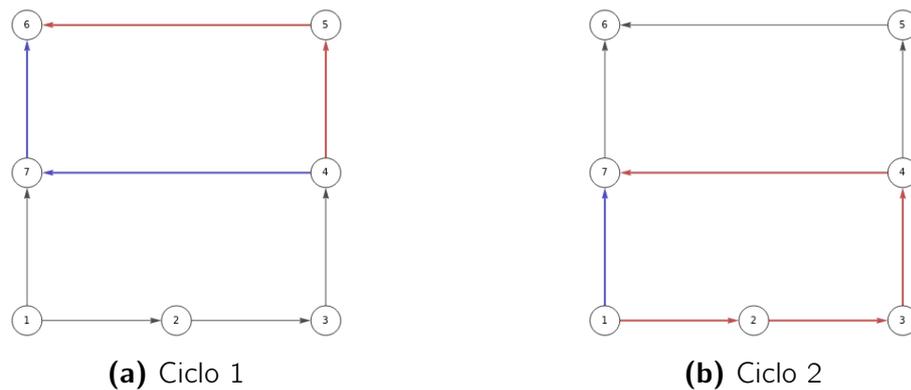
## 2.5. Modelo lineal: caso general sobre una red



**Figura 2.2.** Una red de 7 señales. Se ha considerado cinco arterias. Las flechas indican la dirección de salida elegida. Los números en los segmentos de arteria representan las distancias (en metros) entre señales.

Para la implementación de este caso, se considero el modelo 1.4.2 descrito en la sección 1.4.4. La simulación de este caso se basó en el ejemplo del caso de restricciones de ciclo ubicado en (Little, 1966). Se consideró la disposición de arterias (o calles) mostrada en la figura 2.2.

La base de ciclo considerada en esta simulación es estrictamente fundamental, como se muestra en la figura 2.3. A partir de aquí nos referiremos a las bases “estrictamente fundamentales” simplemente como “fundamentales”.



**Figura 2.3.** Base de ciclo fundamental conformada por dos ciclos. El ciclo 1: (7, 4, 5, 6) y el ciclo 2: (7, 4, 3, 2, 1).

Se consideraron cinco arterias. El orden de estas arterias es el que se muestra en la tabla 2.1.

Clasificación de arterias	
Arteria 1:	del nodo 1 al nodo 3.
Arteria 2:	del nodo 4 al nodo 7.
Arteria 3:	del nodo 5 al nodo 6.
Arteria 4:	del nodo 1 al nodo 6.
Arteria 5:	del nodo 3 al nodo 5.

**Tabla 2.1.** Arterias etiquetadas en función de los nodos.

Por tanto el conjunto de nodos de intersección entre arterias para esta simulación es {1, 3, 4, 5, 6, 7}.

El tamaño de luces rojas en cada arteria son los mismos para la dirección de entrada y salida. Los rojos considerados en cada arteria se muestran en la tabla 2.2.

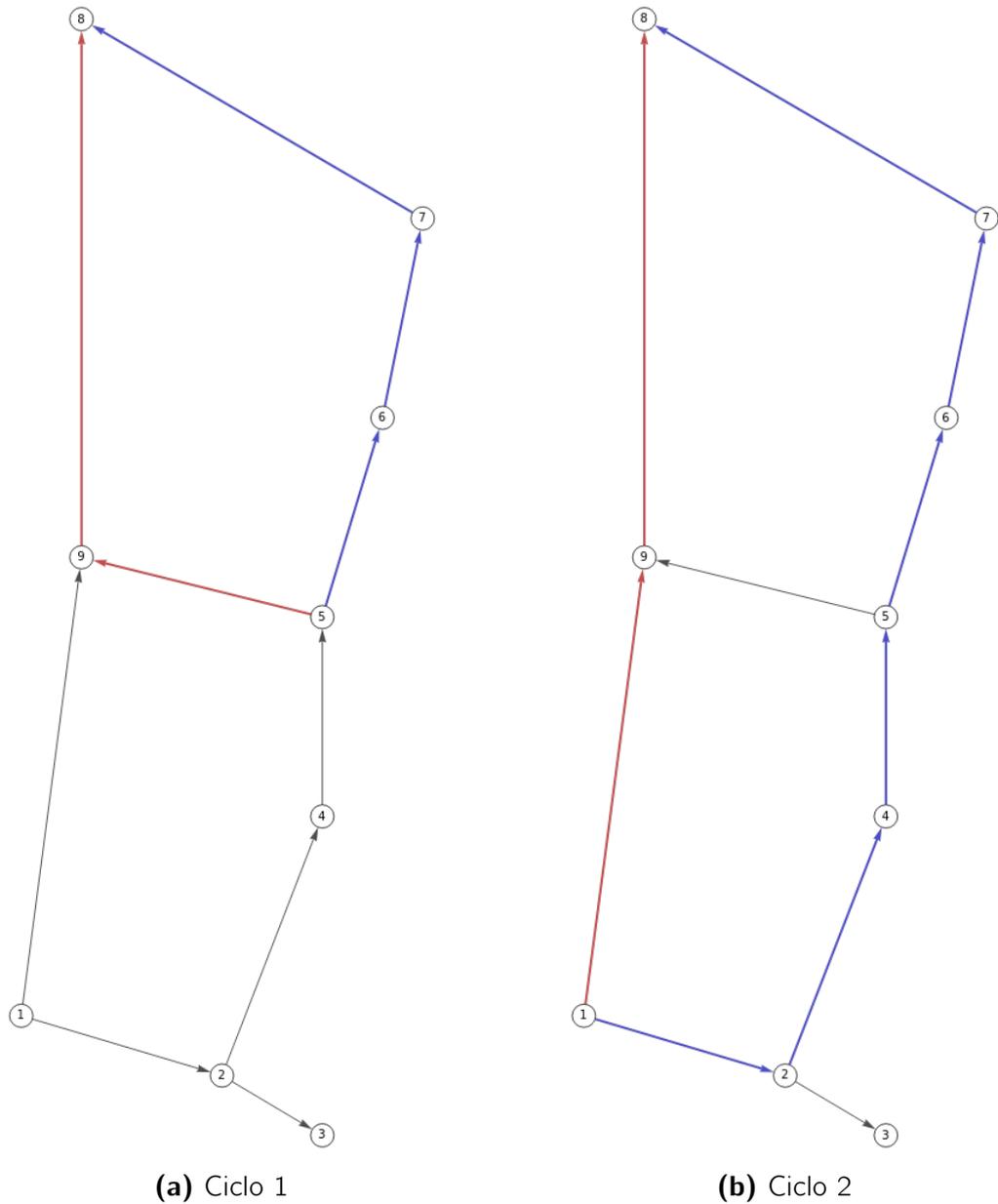
Arteria	Tamaño de luces rojas (ciclos)
1	0,47, 0,40, 0,40
2	0,47, 0,48
3	0,42, 0,40
4	0,40, 0,40, 0,42
5	0,40, 0,40, 0,42

**Tabla 2.2.** Duración en ciclos de las luces rojas por arteria. Las señales están ordenadas por aparición en la dirección de salida de cada arteria.

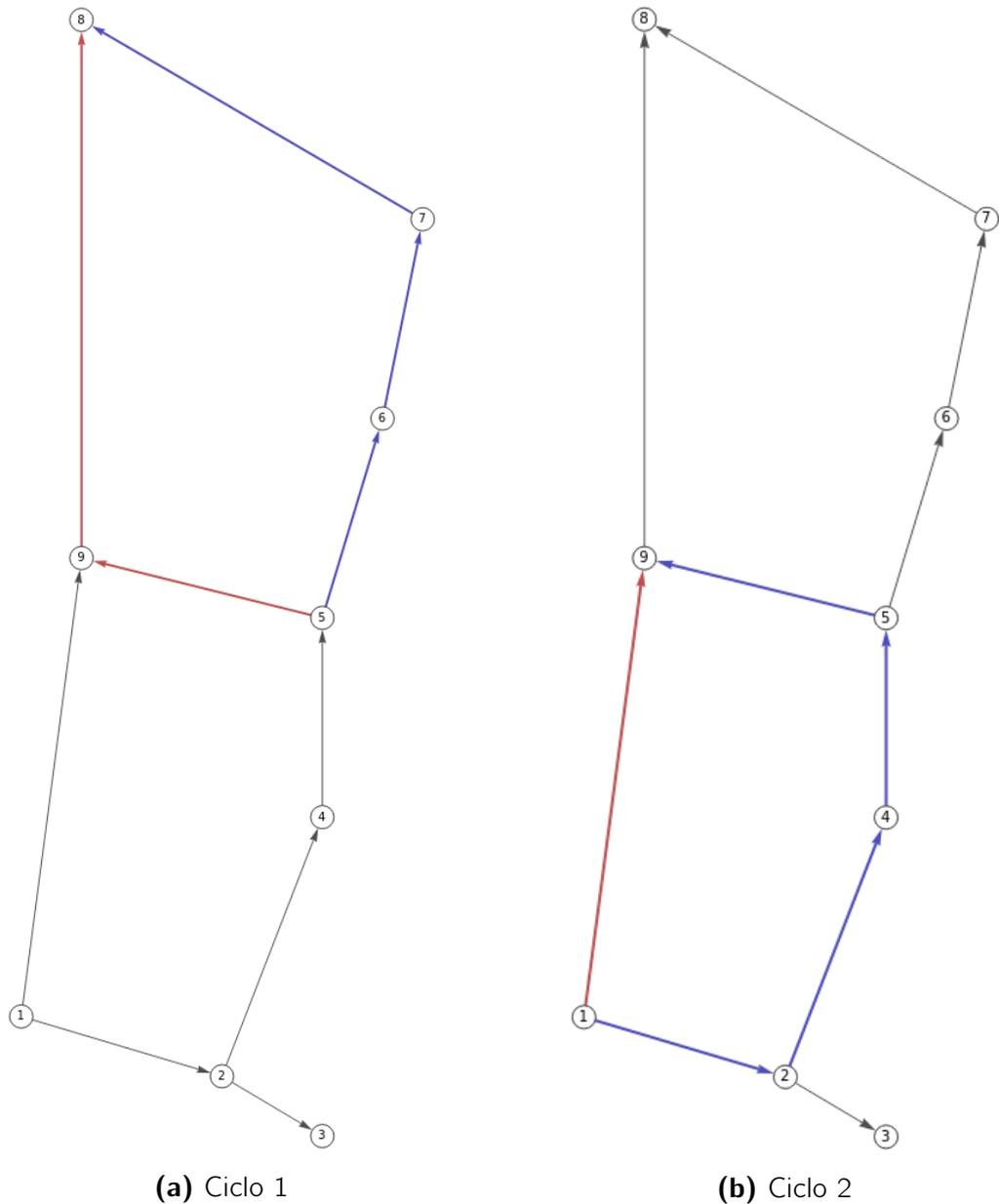


En este caso se consideró el sector de Guayaquil limitado por las avenidas *Francisco de Orellana*, *Rodolfo Baquerizo Nazur*, *Presidente José Luis Tamayo Terán*, *Teodoro Alvarado Oleas* e *Isidro Ayora*, véase la figura 2.4.

Para este caso, se implementó la formulación MAXBAND descrita en el modelo 1.4.2. En esta simulación se consideraron dos bases de ciclos fundamentales, la *fundamental uno* (figura 2.5) y *fundamental dos* (figura 2.6).



**Figura 2.5.** Base de ciclo conformada por dos ciclos. El ciclo 1: (9, 5, 6, 7, 8) y el ciclo 2: (1, 2, 4, 5, 6, 7, 8, 9).



**Figura 2.6.** Base de ciclo conformada por dos ciclos. El ciclo 1: (9, 5, 6, 7, 8) y el ciclo 2: (1, 2, 4, 5, 9).

**Observación 5.** Las restricciones de ciclos consideradas en este problema solo dependen del conjunto de nodos 1, 2, 4, 5, 6, 7, 8, 9. Sin embargo, esto no representa problema alguno para la implementación de la formulación MAXBAND, pues la arista (2, 3) se sincroniza de manera independiente a estas restricciones.

Se consideraron cinco arterias <sup>2</sup>. El orden de estas arterias es el que se muestra en la tabla 2.3.

<sup>2</sup>Arteria 1: Rodolfo Baquerizo Nazur, Arteria 2: José Luis Tamayo Terán, Arteria 3: Teodoro Alvarado Oleas, Arteria 4: Francisco de Orellana y Arteria 5: Isidro Ayora

Clasificación de arterias
Arteria 1: del nodo 1 al nodo 3.
Arteria 2: del nodo 5 al nodo 9.
Arteria 3: del nodo 7 al nodo 8.
Arteria 4: del nodo 1 al nodo 8.
Arteria 5: del nodo 2 al nodo 7.

**Tabla 2.3.** Arterias etiquetadas en función de los nodos.

Por tanto el conjunto de nodos de intersección entre arterias para esta simulación es  $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ .

El tamaño de luces rojas en cada arteria son los mismos para la dirección de entrada y salida. Los rojos considerados en cada arteria se muestran en la tabla 2.4.

Arteria	Tamaño de luces rojas (ciclos)
1	0,47, 0,40, 0,40
2	0,47, 0,48
3	0,42, 0,40
4	0,40, 0,40, 0,42
5	0,40, 0,40, 0,42, 0,40, 0,42

**Tabla 2.4.** Duración en ciclos de las luces rojas por arteria. Las señales están ordenadas por aparición en la dirección de salida de cada arteria.

Así mismo, como en la sección 2.5, el tiempo (en ciclos) de las fases de giro a la izquierda (de entrada y salida) para cada arteria se consideró como el 30% de los rojos de cada señal, en este caso,  $l_{ai} = \bar{l}_{ai} = 0,3 \cdot r_{ai} = 0,3 \cdot \bar{r}_{ai}$ . Los límites de velocidad superior e inferior entre las señales de cada arteria se establecieron en  $17,9 \text{ m/s}$  y  $13,4 \text{ m/s}$ , respectivamente. Así mismo, el límite superior e inferior del cambio de velocidad entre secciones de arteria para cada arteria fue de  $0,0121 \text{ s/m}$  y  $-0,0121 \text{ s/m}$ , respectivamente. El límite inferior y superior del periodo de la luz verde y roja fue de 55 y 75 segundos, respectivamente.

Finalmente, se ejecutó el programa cinco veces para cada base de ciclo considerada en esta simulación. Esto con el fin de comparar el impacto de las bases de ciclos en el rendimiento computacional del programa.

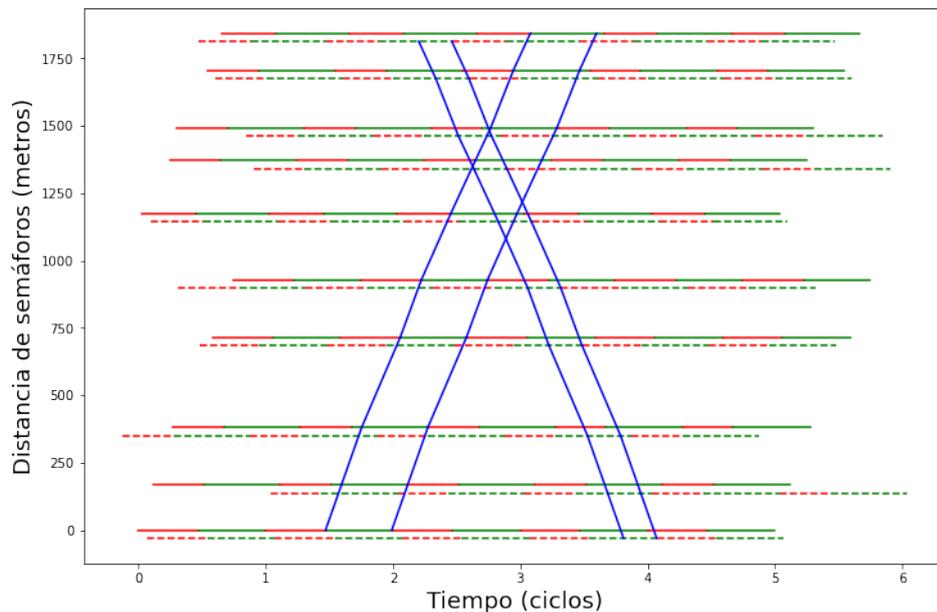
# Capítulo 3

## Resultados de las simulaciones

En este capítulo se presentan los resultados obtenidos de los casos expuestos en las secciones 2.2, 2.3, 2.4, 2.5 y 2.6.

### 3.1. Sobre los diagramas espacio-tiempo

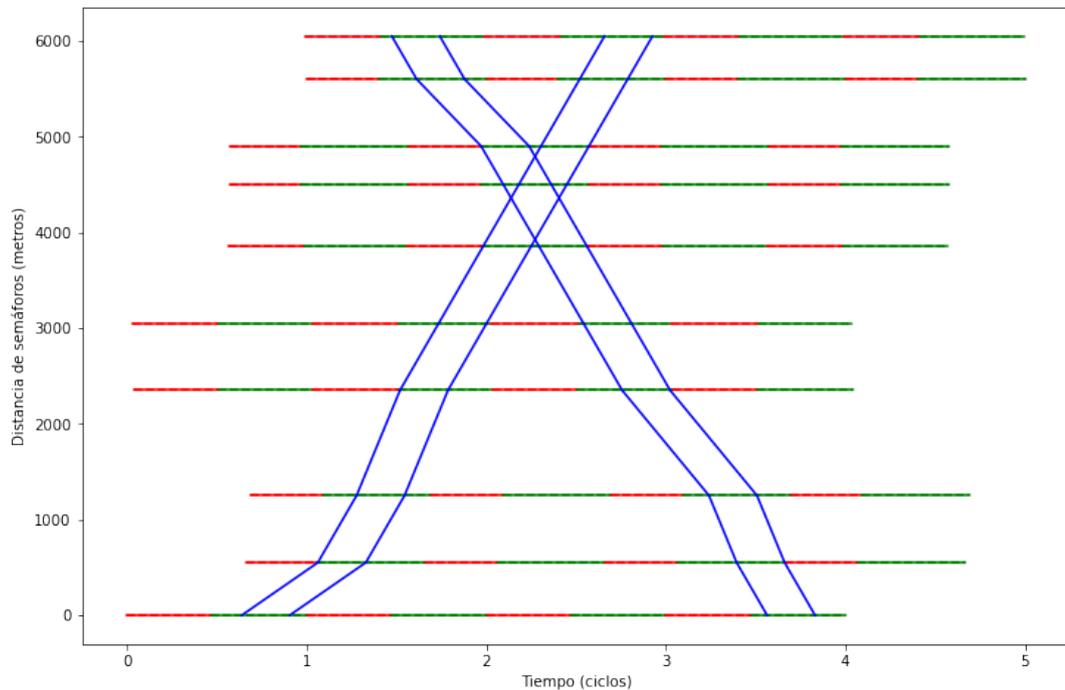
Para la visualización de los resultados se ha implementado diagramas de espacio-tiempo, como el que se muestra en la siguiente figura:



**Figura 3.1.** Las líneas horizontales rojas y verdes en el diagrama espacio-tiempo representan los tiempos de luces rojas y verdes para cada señal a lo largo del tiempo. Un ciclo o periodo es la suma de una luz roja y una luz verde. Las líneas horizontales punteadas representan las luces rojas y verdes de la dirección de entrada o venida. En caso que no existan giros a la izquierda, las líneas continuas y punteadas se solapan. Las líneas azules paralelas representan el ancho del bandwidth. Aquellas con pendiente positiva representan la dirección de ida, y aquellas con pendiente negativa, la dirección de venida.

### 3.2. Resultado del modelo lineal: caso arterial 1

Se obtuvo el siguiente diagrama:



**Figura 3.2.** Diagrama espacio-tiempo de sincronización para el caso arterial 1.

Fijando al primer semáforo como señal crítica<sup>1</sup> se obtuvieron los siguientes valores para la sincronización:

		Sincronización									
h		1	2	3	4	5	6	7	8	9	10
$\theta(1, h)$		0	0.63	0.66	0.03	0	0.54	0.54	0.54	0.96	0.96

**Tabla 3.1.** Valores de  $\theta(1, h)$  para  $h = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ . Los valores fueron redondeados a dos cifras decimales.

Respecto a los valores  $w_i$ , que indican el inicio del bandwidth, se obtuvo:

		Valores de $w_i$									
i		1	2	3	4	5	6	7	8	9	10
$w_i$		0.17	0	0.18	0.01	0.22	0	0.21	0.33	0.12	0.25

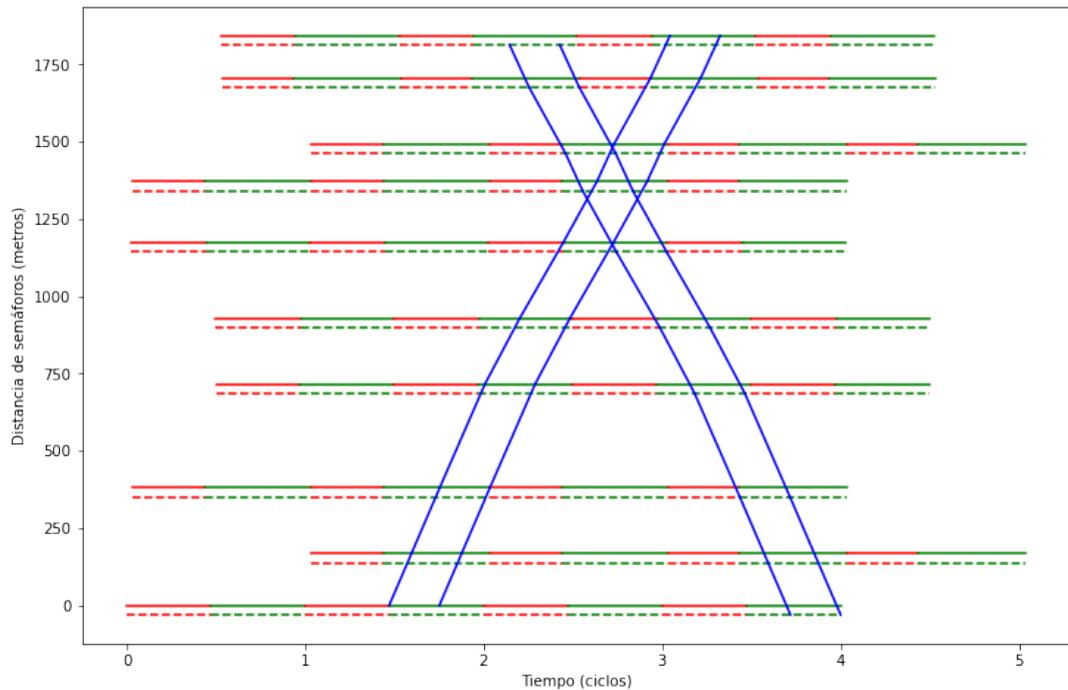
**Tabla 3.2.** Valores de  $w_i$  para cada señal  $i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ . Los valores fueron redondeados a dos cifras decimales. En este caso  $w_i = \bar{w}_i$  para todo  $i$ .

<sup>1</sup>Semáforo seleccionado como origen para sincronizar las otras señales.

Finalmente, el valor del bandwidth resultante, para ambas direcciones (entrada y salida), fue de 0,27 ciclos. Esto quiere decir que el bandwidth tiene una duración de 17,55 segundos.

### 3.3. Resultado del modelo lineal: caso arterial 2

Se obtuvo el siguiente diagrama:



**Figura 3.3.** Diagrama espacio-tiempo de sincronización para el caso arterial 2.

Fijando al primer semáforo como señal crítica se obtuvieron los siguientes valores para la sincronización:

		Sincronización									
h		1	2	3	4	5	6	7	8	9	10
$\theta(1, h)$		0	0	0	0.5	0.5	0	0	0	0	0.5

**Tabla 3.3.** Valores de  $\theta(1, h)$  para  $h = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ . Para las señales = 6, 7, 8 los valores fueron de 1, sin embargo, esto es equivalente a poner 0 si desplazamos un ciclo hacia la izquierda.

Respecto a los valores  $w_j$ , que indican el inicio del bandwidth, se obtuvo:

		Valores de $w_i$									
$i$		1	2	3	4	5	6	7	8	9	10
$w_i$		0	0.2	0.3	0	0.2	0	0.2	0.3	0	0.1

**Tabla 3.4.** Valores de  $w_i$  para cada señal  $i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ . En este caso  $w_i = \bar{w}_i$  para todo  $i$ .

Las velocidades  $v_{i,i+1}$  entre señales fueron:

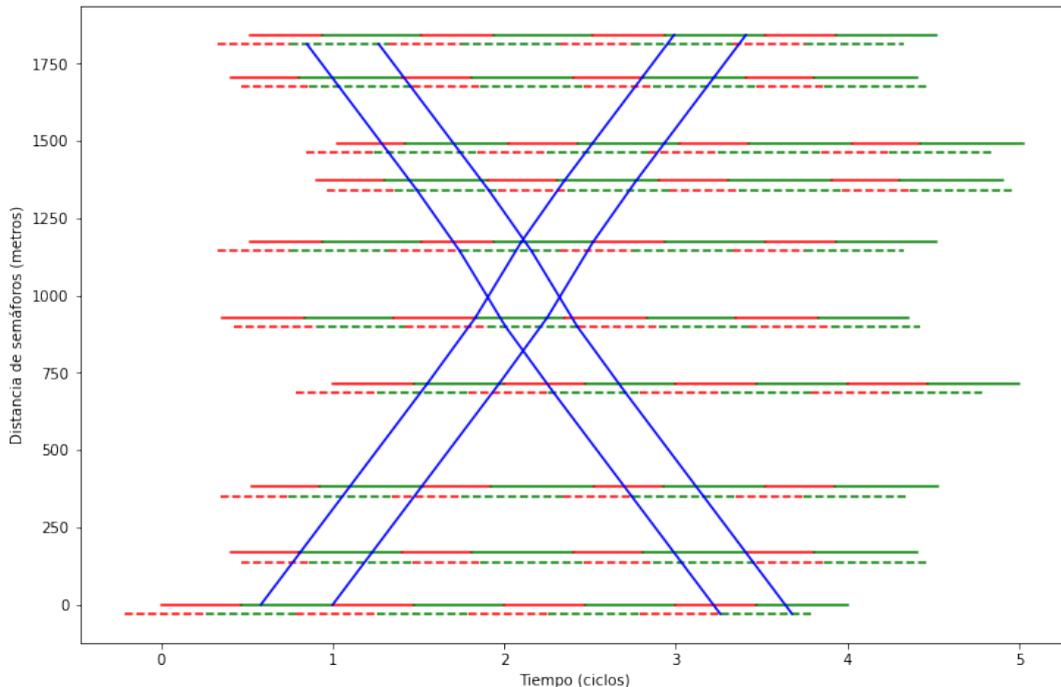
		Velocidades entre señales (m/s)								
$i$		1	2	3	4	5	6	7	8	9
$v_{i,i+1}$		17.9	17.9	17.6	14.5	13.4	13.9	16.8	13.9	16.8

**Tabla 3.5.** Valores obtenidos de las velocidades entre señales a los que se debe circular para aprovechar el bandwidth. Los valores fueron redondeados a una cifra decimal. En este caso,  $v_{i,i+1} = \bar{v}_{i,i+1}$ .

Finalmente, el valor de longitud de periodo para la luz roja y verde fue de 75 segundos y el bandwidth resultante, para ambas direcciones (entrada y salida), fue de 0,28 ciclos. Esto quiere decir que el bandwidth tiene una duración de 21 segundos.

### 3.4. Resultado del modelo lineal: caso arterial 3

Se obtuvo el siguiente diagrama:



**Figura 3.4.** Diagrama espacio-tiempo de sincronización para el caso arterial 3.

Fijando al primer semáforo como señal crítica se obtuvieron los siguientes valores para la sincronización:

		Sincronización									
$h$		1	2	3	4	5	6	7	8	9	10
$\theta(1, h)$		0	0.37	0.49	0	0.36	0.49	0.13	0.01	0.37	0.49

**Tabla 3.6.** Valores de  $\theta(1, h)$  para  $h = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ . Los valores mayores a 0,5, se movieron un ciclo hacia la izquierda.

Respecto a los valores  $w_i$ , que indican el inicio del bandwidth, se obtuvo:

		Valores de $w_i$									
$i$		1	2	3	4	5	6	7	8	9	10
$w_i$		0.1	0	0.2	0.1	0	0.2	0	0.1	0	0.1

**Tabla 3.7.** Valores de  $w_i$  para cada señal  $i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ . En este caso  $w_i = \bar{w}_i$  para todo  $i$ .

Las velocidades  $v_{i,i+1}$  entre señales fueron:

		Velocidades entre señales ( $m/s$ )									
$i$		1	2	3	4	5	6	7	8	9	
$v_{i,i+1}$		13.4	13.4	13.4	13.8	16.6	14.3	13.4	13.4	13.4	

**Tabla 3.8.** Valores obtenidos de las velocidades entre señales a los que se debe circular para aprovechar el bandwidth. Los valores fueron redondeados a una cifra decimal. En este caso,  $v_{i,i+1} = \bar{v}_{i,i+1}$ .

Respecto a los giros hacia la izquierda, se obtuvo la siguiente configuración:

		Fases de giros hacia la izquierda									
$i$		1	2	3	4	5	6	7	8	9	10
$\delta_i$		1	0	1	1	0	1	0	1	0	1
$\bar{\delta}_i$		0	1	0	0	1	0	1	0	1	0
Patrón		2	1	2	2	1	2	1	2	1	2

**Tabla 3.9.** Fases de giro a la izquierda para cada señal  $i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ , en función de las variables binarias  $\delta_i$  y  $\bar{\delta}_i$ . Véase la figura 1.2 para detalles sobre el patrón de giro.

Finalmente, el valor de longitud de periodo para la luz roja y verde fue de 55 segundos y el bandwidth resultante, para ambas direcciones (entrada y salida), fue de 0,42 ciclos. Esto quiere decir que el bandwidth tiene una duración de 23,1 segundos.

### 3.5. Resultado del modelo lineal: caso sobre una red

Fijando al primer semáforo como señal crítica para cada arteria, se obtuvieron los siguientes valores para la sincronización:

(a) Dirección de salida					(b) Dirección de entrada				
Sincronización					Sincronización				
Arteria	$h$	1	2	3	Arteria	$h$	1	2	3
1	$\theta(1, h)$	0	0.13	0.43	1	$\theta(1, h)$	0	0.13	0.59
2	$\theta(1, h)$	0	0.56		2	$\theta(1, h)$	0	0.59	
3	$\theta(1, h)$	0	0.53		3	$\theta(1, h)$	0	0.59	
4	$\theta(1, h)$	0	0.12	0.5	4	$\theta(1, h)$	0	0.12	0.49
5	$\theta(1, h)$	0	0.12	0.5	5	$\theta(1, h)$	0	0.12	0.49

**Tabla 3.10.** Valores de  $\theta(1, h)$  y  $\bar{\theta}(1, h)$  para cada arteria, véase las subtablas 3.10a y 3.10b, respectivamente. El número máximo de señales en una arteria fue de tres. Los espacios vacíos indican que no existe valor de  $\theta(1, h)$  o  $\bar{\theta}(1, h)$  porque tal señal no existe.

Respecto a los valores  $w_i$  para cada arteria, se obtuvo:

(a) Dirección de salida					(b) Dirección de entrada				
Valores de $w_i$					Valores de $w_i$				
Arteria	$i$	1	2	3	Arteria	$i$	1	2	3
1	$w_i$	0	0.11	0.11	1	$w_i$	0	0.11	0
2	$w_i$	0	0		2	$w_i$	0	0	
3	$w_i$	0	0.02		3	$w_i$	0	0.02	
4	$w_i$	0	0.05	0	4	$w_i$	0	0.05	0
5	$w_i$	0	0.05	0	5	$w_i$	0	0.05	0

**Tabla 3.11.** Valores de  $w_i$  y  $\bar{w}_i$  para cada arteria, véase las subtablas 3.11a y 3.11b, respectivamente. El número máximo de señales en una arteria fue de tres. Los espacios vacíos indican que no existe valor de  $w_i$  o  $\bar{w}_i$  porque tal señal no existe.

Las velocidades  $v_{i,i+1}$  entre señales, para cada arteria, fueron:

(a) Dirección de salida					(b) Dirección de entrada				
Velocidades entre señales ( $m/s$ )					Velocidades entre señales ( $m/s$ )				
Arteria	$i$	1	2		Arteria	$i$	1	2	
1	$v_{i,i+1}$	17.9	17.9		1	$v_{i,i+1}$	17.9	15.48	
2	$v_{i,i+1}$	16.08			2	$v_{i,i+1}$	15.28		
3	$v_{i,i+1}$	16.68			3	$v_{i,i+1}$	15.12		
4	$v_{i,i+1}$	15.99	13.4		4	$v_{i,i+1}$	15.99	13.4	
5	$v_{i,i+1}$	15.99	13.4		5	$v_{i,i+1}$	15.99	13.4	

**Tabla 3.12.** Valores de  $v_{i,i+1}$  y  $\bar{v}_{i,i+1}$  para cada arteria, véase las subtablas 3.12a y 3.12b, respectivamente. Espacios vacíos indican que no existe valor de  $v_{i,i+1}$  o  $\bar{v}_{i,i+1}$ .

Respecto a los giros hacia la izquierda, para cada arteria, se obtuvo la siguiente configuración:

Fases de giros hacia la izquierda				
Arteria	$i$	1	2	3
1	$\delta_i$	0	1	0
	$\bar{\delta}_i$	1	0	1
	Patrón	1	2	1
2	$\delta_i$	0	1	
	$\bar{\delta}_i$	0	0	
	Patrón	3	2	
3	$\delta_i$	0	1	
	$\bar{\delta}_i$	0	0	
	Patrón	3	2	
4	$\delta_i$	0	1	0
	$\bar{\delta}_i$	1	0	1
	Patrón	1	2	1
5	$\delta_i$	0	1	0
	$\bar{\delta}_i$	1	0	1
	Patrón	1	2	1

**Tabla 3.13.** Fases de giro a la izquierda para cada señal de cada arteria, en función de las variables binarias  $\delta_i$  y  $\bar{\delta}_i$ . Véase la figura 1.2 para detalles sobre el patrón de giro. Los espacios vacíos indican valores que no existen.

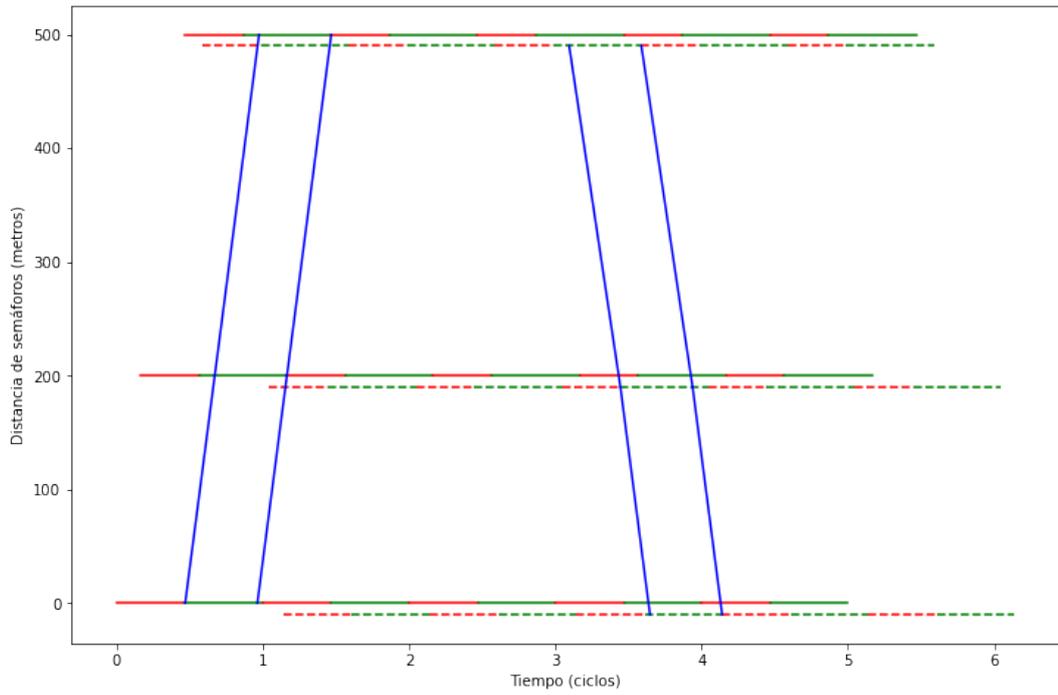
Finalmente, el tiempo de periodo de luces rojas y verde, para cada arteria, fue de 55, 30 segundos. Se obtuvieron, para cada arteria, los siguiente valores de bandwidth:

Valores de bandwidth (ciclos)					
arteria	1	2	3	4	5
$b$	0.49	0.52	0.58	0.55	0.55

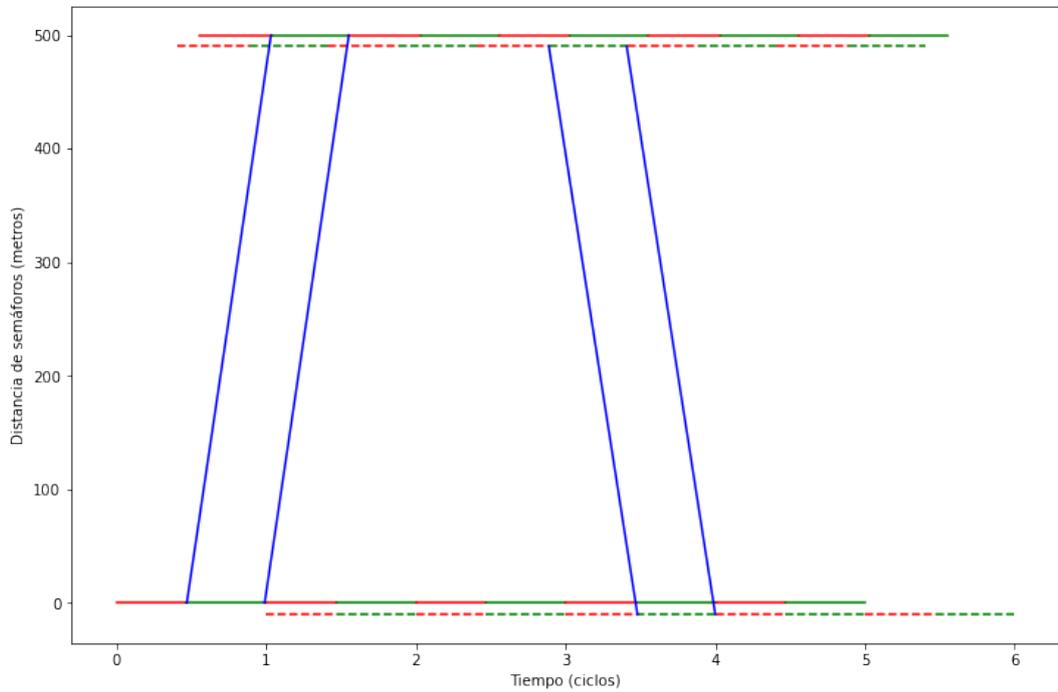
**Tabla 3.14.** Valores del bandwidth para cada arteria en dirección de entrada y salida. Se supone que  $b = \bar{b}$ .

Esto es que, la duración del bandwidth para cada arteria es de 27,1, 28,76, 32,07, 30,42 y 30,42 segundos, respectivamente. Así, la arteria 3 tuvo el bandwidth de mayor duración.

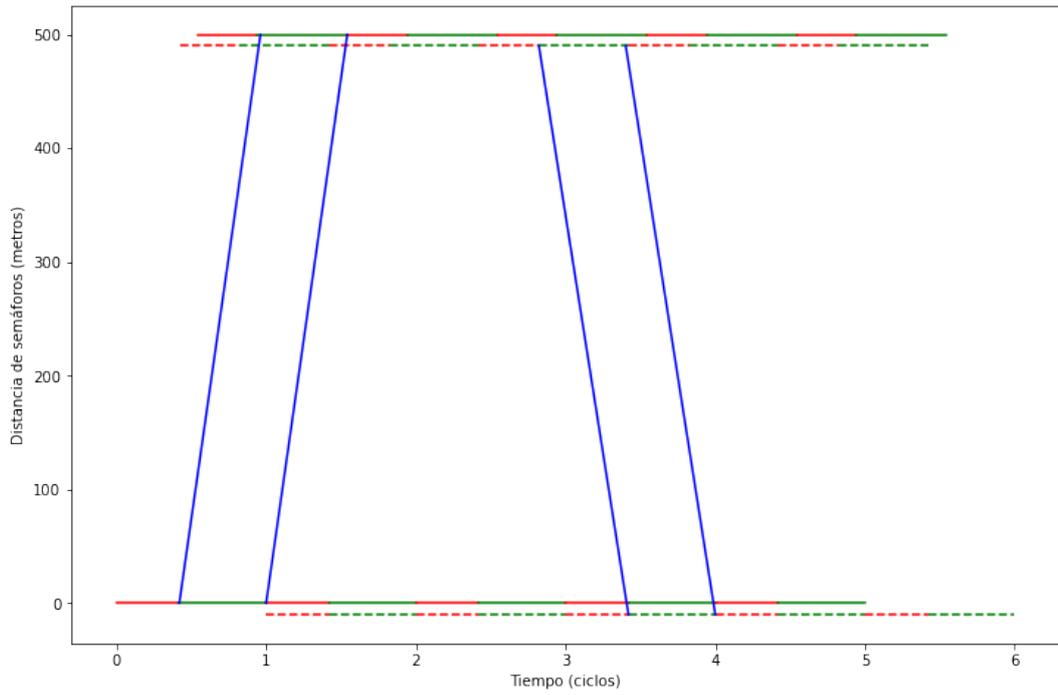
Se obtuvieron los siguientes diagramas:



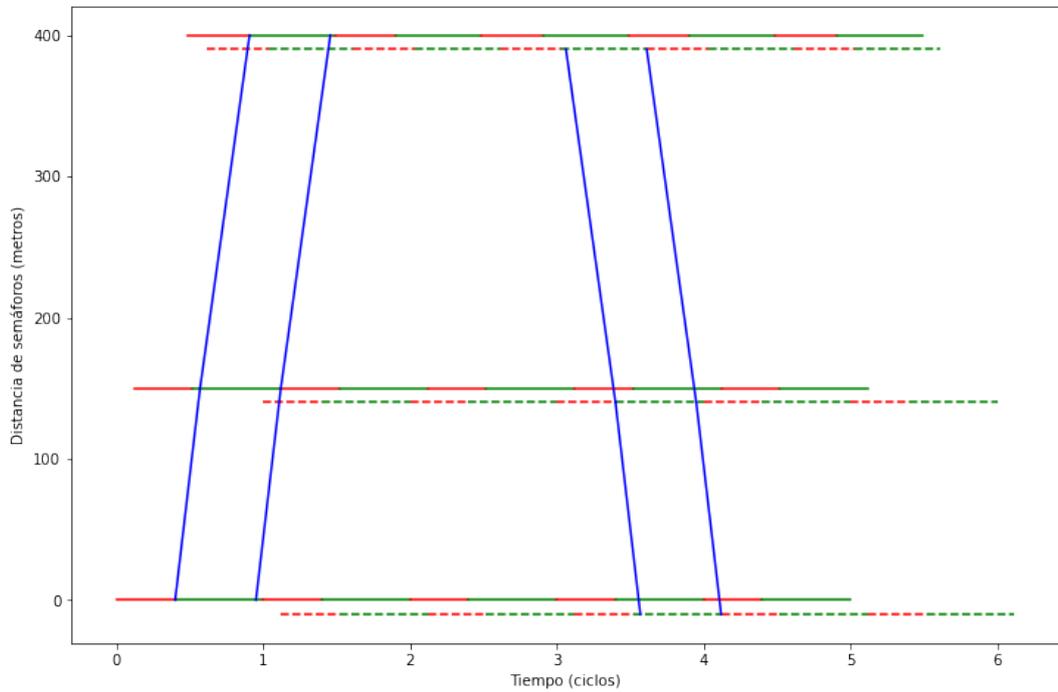
**Figura 3.5.** Diagrama espacio-tiempo de sincronización para el caso sobre una red: arteria 1.



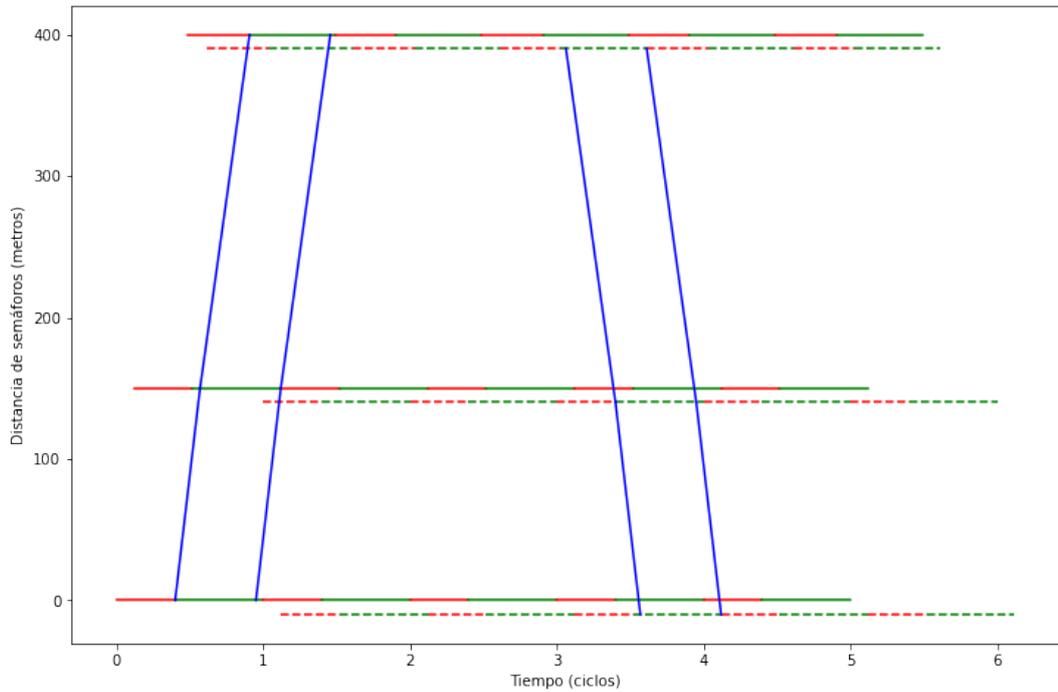
**Figura 3.6.** Diagrama espacio-tiempo de sincronización para el caso sobre una red: arteria 2.



**Figura 3.7.** Diagrama espacio-tiempo de sincronización para el caso sobre una red: arteria 3.



**Figura 3.8.** Diagrama espacio-tiempo de sincronización para el caso sobre una red: arteria 4.



**Figura 3.9.** Diagrama espacio-tiempo de sincronización para el caso sobre una red: arteria 5.

### 3.6. Resultado del modelo lineal: caso real

Fijando al primer semáforo como señal crítica para cada arteria, se obtuvieron los siguientes valores para la sincronización:

(a) Dirección de salida						(b) Dirección de entrada							
		Sincronización							Sincronización				
Arteria	$h$	1	2	3	4	5	Arteria	$h$	1	2	3	4	5
1	$\theta_{1,h}$	0	0.51	0.93			1	$\theta_{1,h}$	0	0.35	0.81		
2	$\theta_{1,h}$	0	0.66				2	$\theta_{1,h}$	0	0.48			
3	$\theta_{1,h}$	0	0.63				3	$\theta_{1,h}$	0	0.12			
4	$\theta_{1,h}$	0	0.30	0.80			4	$\theta_{1,h}$	0	0.58	0.96		
5	$\theta_{1,h}$	0	0.97	0.65	0.53	0.65	5	$\theta_{1,h}$	0	0.79	0.36	0.23	0.36

**Tabla 3.15.** Valores de  $\theta_{1,h} := \theta(1, h)$  y  $\bar{\theta}_{1,h} := \bar{\theta}(1, h)$  para cada arteria, véase las subtablas 3.15a y 3.15b, respectivamente. El número máximo de señales en una arteria fue de tres. Los espacios vacíos indican que no existe valor de  $\theta(1, h)$  o  $\bar{\theta}(1, h)$  porque tal señal no existe.

Respecto a los valores  $w_i$  para cada arteria, se obtuvo:

(a) Dirección de salida						(b) Dirección de entrada							
Valores de $w_i$						Valores de $w_i$							
Arteria	$i$	1	2	3	4	5	Arteria	$i$	1	2	3	4	5
1	$w_i$	0	0	0			1	$w_i$	0	0.07	0		
2	$w_i$	0	0.01				2	$w_i$	0	0			
3	$w_i$	0	0				3	$w_i$	0	0			
4	$w_i$	0	0.02	0			4	$w_i$	0	0	0		
5	$w_i$	0	0.02	0	0	0	5	$w_i$	0	0.02	0	0	0

**Tabla 3.16.** Valores de  $w_i$  y  $\bar{w}_i$  para cada arteria, véase las subtablas 3.16a y 3.16b, respectivamente. El número máximo de señales en una arteria fue de tres. Los espacios vacíos indican que no existe valor de  $w_i$  o  $\bar{w}_i$  porque tal señal no existe.

Las velocidades  $v_{i,i+1}$  entre señales, para cada arteria, fueron:

(a) Dirección de salida						(b) Dirección de entrada					
Velocidades entre señales (m/s)						Velocidades entre señales (m/s)					
Arteria	$i$	1	2	3	4	Arteria	$i$	1	2	3	4
1	$v_{i,i+1}$	13.4	13.4			1	$v_{i,i+1}$	16.63	14.16		
2	$v_{i,i+1}$	13.4				2	$v_{i,i+1}$	17.89			
3	$v_{i,i+1}$	17.67				3	$v_{i,i+1}$	13.59			
4	$v_{i,i+1}$	16.84	13.99			4	$v_{i,i+1}$	14.16	14.87		
5	$v_{i,i+1}$	13.4	15.05	14.71	17.9	5	$v_{i,i+1}$	16.61	17.9	17.71	17.9

**Tabla 3.17.** Valores de  $v_{i,i+1}$  y  $\bar{v}_{i,i+1}$  para cada arteria, véase las subtablas 3.17a y 3.17b, respectivamente. Espacios vacíos indican que no existe valor de  $v_{i,i+1}$  o  $\bar{v}_{i,i+1}$ .

Respecto a los giros hacia la izquierda, para cada arteria, se obtuvo la siguiente configuración:

Fases de giros hacia la izquierda						
Arteria	$i$	1	2	3	4	5
1	$\delta_i$	1	0	0		
	$\bar{\delta}_i$	0	0	1		
	Patrón	2	3	1		
2	$\delta_i$	0	1			
	$\bar{\delta}_i$	0	0			
	Patrón	3	2			
3	$\delta_i$	1	0			
	$\bar{\delta}_i$	0	1			
	Patrón	2	1			
4	$\delta_i$	1	0	0		
	$\bar{\delta}_i$	0	0	1		
	Patrón	2	3	1		
5	$\delta_i$	1	0	1	0	1
	$\bar{\delta}_i$	0	1	0	1	0
	Patrón	2	1	2	1	2

**Tabla 3.18.** Fases de giro a la izquierda para cada señal de cada arteria, en función de variables binarias  $\delta_i$  y  $\bar{\delta}_i$ . Véase 1.2 para detalles sobre el patrón de giro. Espacios vacíos indica que no existe valor.

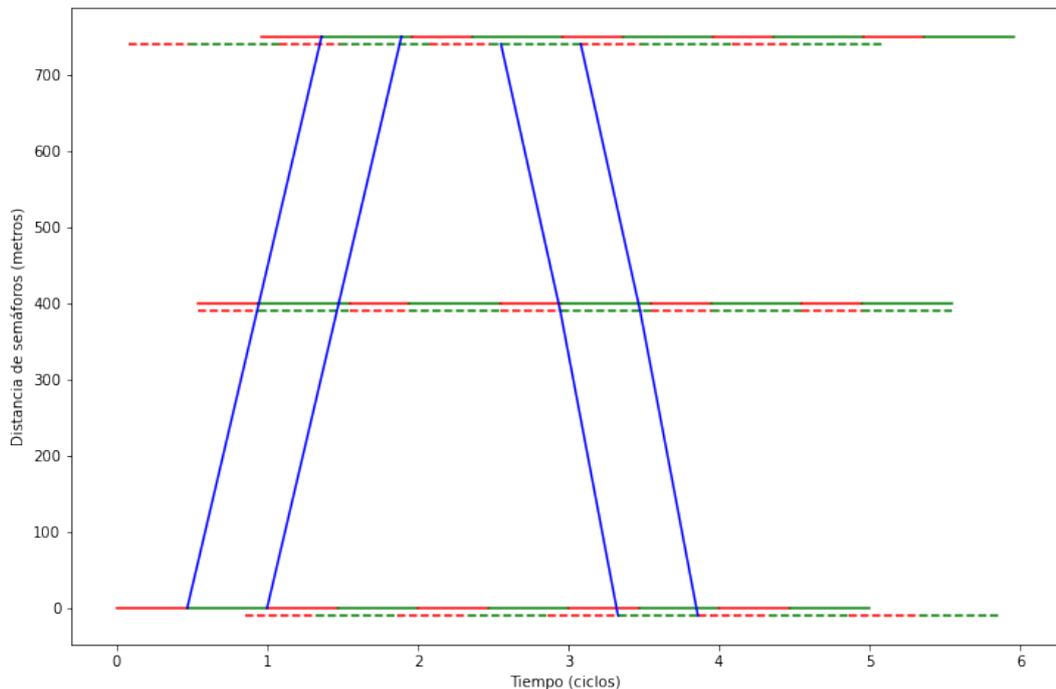
El tiempo de periodo de luces rojas y verde, para cada arteria, fue de 62,75 segundos. Se obtuvieron, para cada arteria, los siguiente valores de bandwith:

Valores de bandwith (ciclos)					
arteria	1	2	3	4	5
$b$	0.53	0.52	0.58	0.58	0.58

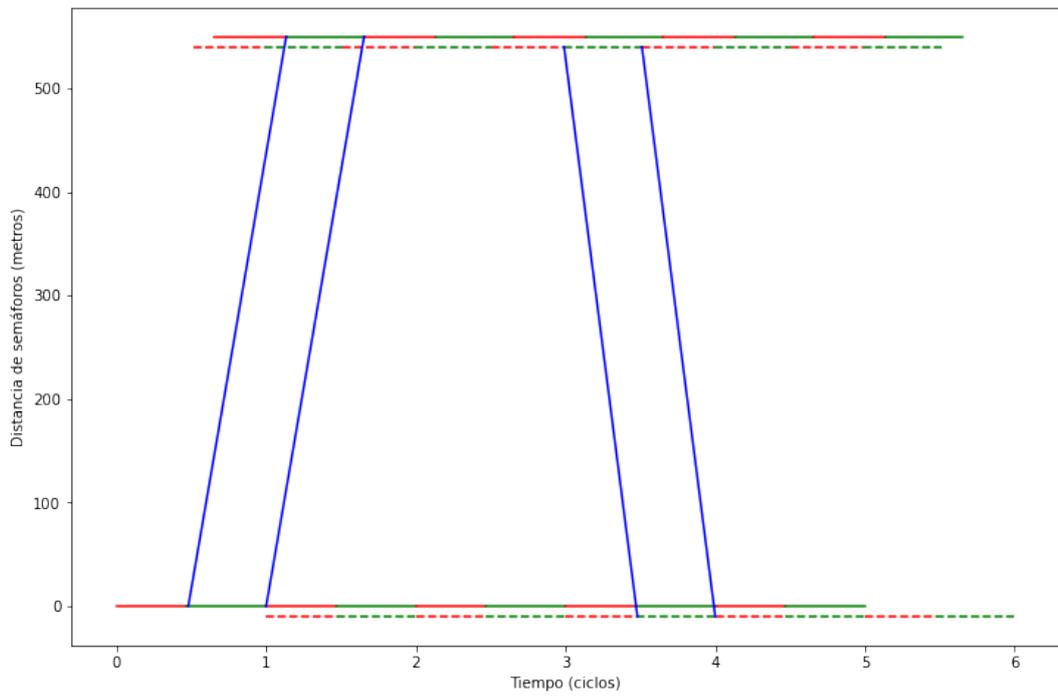
**Tabla 3.19.** Valores del bandwith para cada arteria en dirección de entrada y salida. Se supone que  $b = \bar{b}$ .

Esto es que, la duración del bandwith para cada arteria es de 33,26, 32,63, 36,40, 36,40 y 36,40 segundos, respectivamente. Así, las arterias 3, 4 y 5, tuvieron el bandwith de mayor duración.

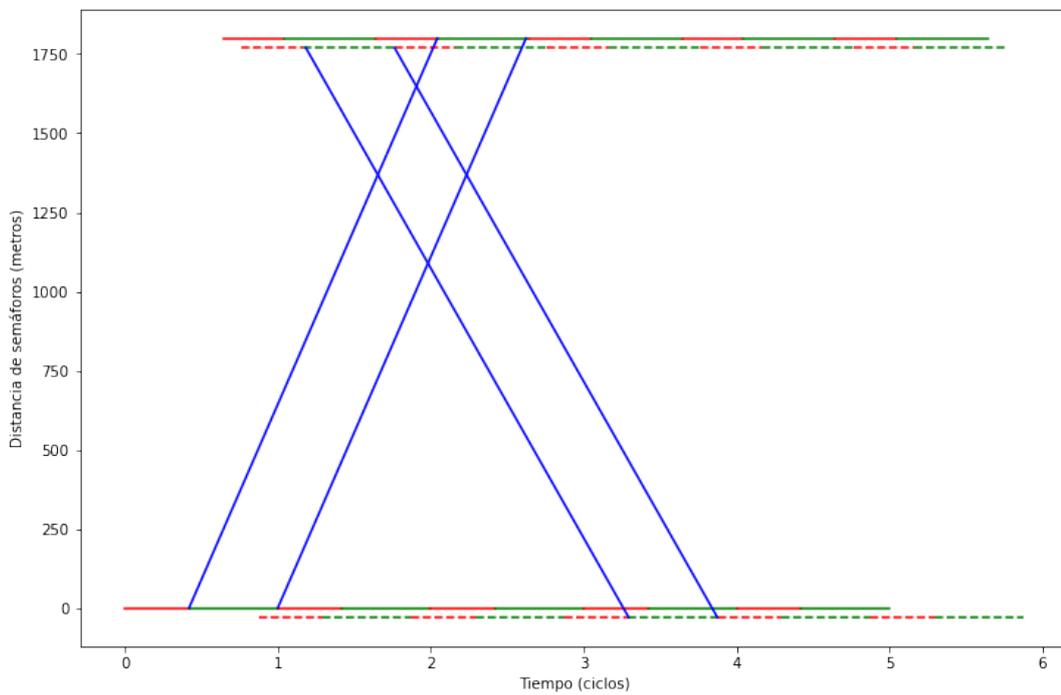
Se obtuvieron los siguientes diagramas:



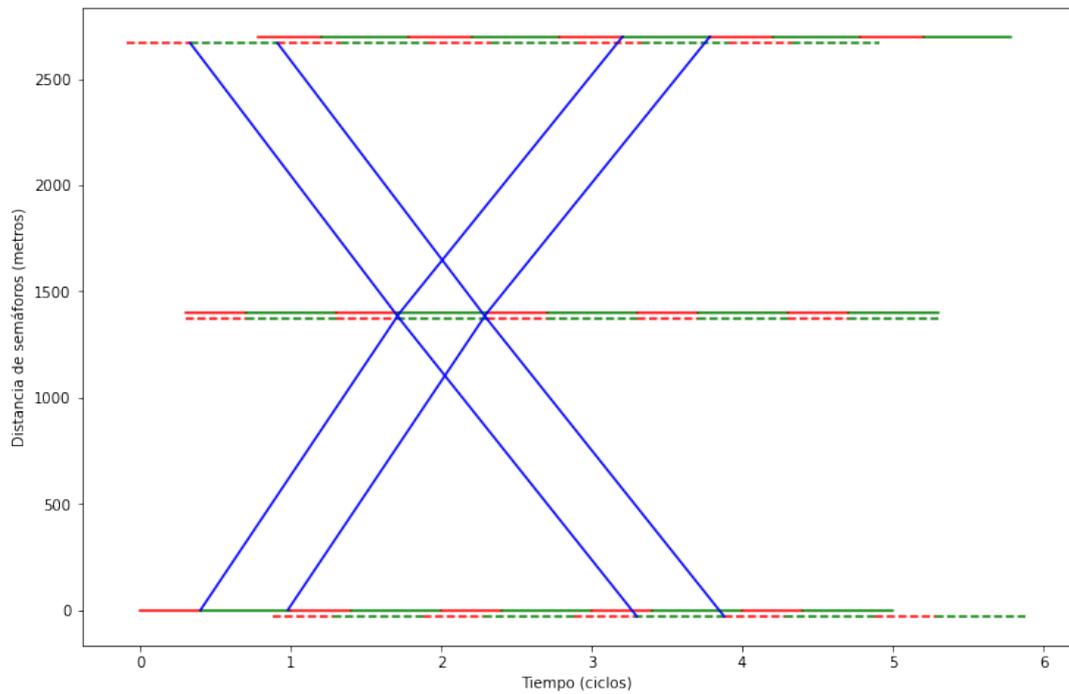
**Figura 3.10.** Diagrama espacio-tiempo de sincronización para el caso real: arteria 1.



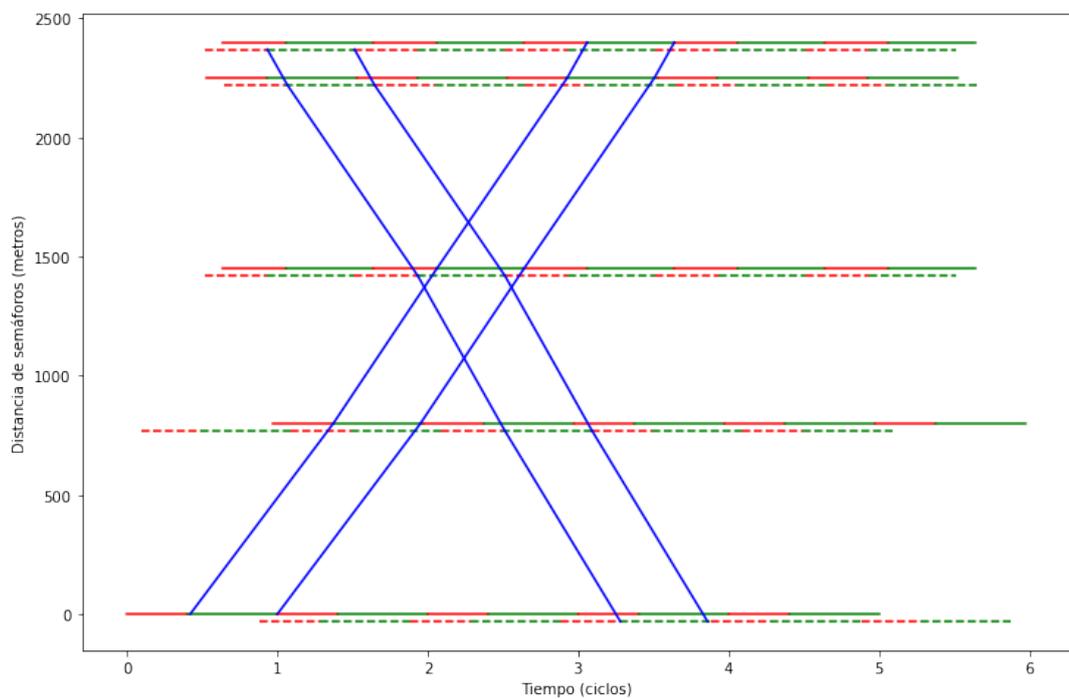
**Figura 3.11.** Diagrama espacio-tiempo de sincronización para el caso real: arteria 2.



**Figura 3.12.** Diagrama espacio-tiempo de sincronización para el caso real: arteria 3.



**Figura 3.13.** Diagrama espacio-tiempo de sincronización para el caso real: arteria 4.



**Figura 3.14.** Diagrama espacio-tiempo de sincronización para el caso real: arteria 5.

**Observación 6.** Tanto para la base fundamental uno como para la base fundamental dos, se obtuvieron los mismos resultados, y por tanto los mismos diagramas, expuestos en esta sección.

Finalmente, los resultados del rendimiento en tiempo de ejecución de las bases de ciclos en el programa, se resumen en la siguiente tabla:

Ejecución	Tiempo de ejecución en segundos	
	Base fundamental 1	Base fundamental 2
1	0.476	0.355
2	0.512	0.404
3	0.507	0.211
4	0.616	0.502
5	0.521	0.261

**Tabla 3.20.** Tiempo de ejecución de la formulación MAXBAND sin considerar tiempos de construcción del modelo a partir de los datos de entrada.

En esta simulación, la base fundamental dos tuvo el menor tiempo promedio de ejecución. De hecho, tuvo un tiempo de ejecución menor en todas las ejecuciones del programa. Se podría pensar que esto se debe a que la base fundamental dos es la base de menor peso<sup>2</sup>, sin embargo, esto se debe estudiar a mayor profundidad bajo otras condiciones.

<sup>2</sup>La base fundamental uno tuvo un peso de 11900 unidades, mientras que la base fundamental dos, tuvo un peso de 8400 unidades

# Capítulo 4

## Conclusiones y recomendaciones

### 4.1. Conclusiones

Con la finalidad de mejorar la sincronización de semáforos de cualquier sector en cualquier ciudad del mundo, se implementó la formulación MAXBAND descrita en el trabajo de Cabezas et al. (2019)), utilizando el paquete Pyomo-Python junto con el solver *CPLEX*. Luego de los procesos metodológicos y resultados descritos en los capítulos 2 y 3, respectivamente, se concluye que:

1. La formulación del modelo MAXBAND es muy versátil puesto que, entre otras cosas, permite considerar giros a la izquierda, velocidades variables, límites de velocidad, factores de tráfico, importancia de carriles (entrada y salida) a través de pesos asignados a los bandwidth (entrada y salida), y restricciones de ciclos. Esto permite modelar muchas situaciones que se presentan en la vida real en relación con la sincronización de semáforos, obteniendo resultados muy favorables, por ejemplo, los resultados que se obtuvieron para el sector seleccionado del norte de Guayaquil en este trabajo.
2. A pesar que los tiempos de las fases de giros hacia la izquierda fueron introducidos como un porcentaje moderado y aleatorio del tiempo de la luz roja, es posible, en función del volumen y capacidad del tráfico, determinar un porcentaje ajustado a la realidad. Para esto, es muy utilizada la teoría de Webster (1958), descrita en la sección 1.4.5. La implementación de esta teoría, en la formulación MAXBAND, permitirá obtener una sincronización óptima de semáforos que esté más ajustada a la realidad inmediata de las calles en estudio.
3. Con base en los experimentos realizados, se pudo observar que, utilizando los mismos parámetros para distintas bases de ciclos sobre una red, se obtienen resultados iguales, véase la sección 3.6. Sin embargo, el desempeño computacional en el tiempo de obtención de resultados de la formulación MAXBAND sobre una red, parece ser inversamente proporcional al peso de la base de ciclos utilizada.
4. Se espera que el enfoque de este trabajo sea motivación para otros autores en el uso de MAXBAND para el caso sobre una red y que las herramientas

computacionales propuestas sean un aporte a este propósito.

## 4.2. Recomendaciones

En función del trabajo realizado, para estudios posteriores se recomienda:

1. Contrastar los resultados obtenidos con simulaciones en programas especializados de tránsito, por ejemplo, *Vissim*. Ver (PTV AG, 2022).
2. Utilizar programas como *Qgis* y *OpenStreetMap* , ver (QGIS Development Team, 2009) y (OpenStreetMap contributors, 2017) respectivamente, para la obtención de los parámetros mínimos para el funcionamiento del modelo MAXBAND, por ejemplo, el número de semáforos y distancia entre ellos, en un determinado sector de estudio.
3. Implementar la teoría de Webster (1958) dentro de la formulación MAXBAND, de tal manera que al ingresar los requerimientos mínimos descritos en el algoritmo 1 , el programa en Python calcule los tiempos de las fases de giros hacia la izquierda que minimicen el tiempo de demora en las intersecciones.
4. Llamar desde python, la base de ciclo que se obtiene desde *Wolfram Mathematica* (Wolfram Research, Inc., 2010), en la implementación de la formulación MAXBAND. Ver anexo A.
5. Estudiar el impacto de diferentes clases de base de ciclos (no solo bases fundamentales) en el rendimiento computacional de la implementación del modelo MAXBAND y su relación con el peso de las bases. Nótese que en este trabajo solo se trabajó con dos bases fundamentales distintas.

# Bibliografía

- [1] Michael L. Bynum et al. *Pyomo—optimization modeling in python*. Third. Vol. 67. Springer Science & Business Media, 2021.
- [2] Xavier Cabezas et al. «A hybrid heuristic approach for traffic light synchronization based on the MAXBAND». En: *Soft Computing Letters* 1 (2019), pág. 100001.
- [3] José Xavier Cabezas García. «Análisis y comparación de modelos de programación matemática para la sincronización de señales de tránsito: estudio sobre una arteria principal y una sección de la red de transporte en guayaquil». En: (2015).
- [4] IBM ILOG Cplex. «V12. 1: User's Manual for CPLEX». En: *International Business Machines Corporation* 46.53 (2009), pág. 157.
- [5] Reinhard Diestel. «Graph theory 3rd ed». En: *Graduate texts in mathematics* 173 (2005), pág. 33.
- [6] Nathan H Gartner et al. «Optimization of traffic signal settings by mixed-integer linear programming: Part I: The network coordination problem». En: *Transportation Science* 9.4 (1975), págs. 321-343.
- [7] INRIX. *Global Traffic Scorecard 2021*. Ene. de 2021. URL: <https://inrix.com/scorecard/>.
- [8] Donald Ervin Knuth et al. *The art of computer programming*. Vol. 3. Addison-Wesley Reading, MA, 1973.
- [9] Christian Liebchen et al. «Classes of cycle bases». En: *Discrete Applied Mathematics* 155.3 (2007), págs. 337-355.
- [10] John DC Little. «The synchronization of traffic signals by mixed-integer linear programming». En: *Operations Research* 14.4 (1966), págs. 568-594.
- [11] John DC Little et al. «MAXBAND: A versatile program for setting signals on arteries and triangular networks». En: (1981).
- [12] Google Maps. <https://www.google.com.ec/maps/@-2.1229163,-79.8933664,15z?hl=en-GB&authuser=0>. Accessed: 2022-8-31. 2022.
- [13] John T Morgan et al. «Synchronizing traffic signals for maximal bandwidth». En: *Operations Research* 12.6 (1964), págs. 896-912.
- [14] OpenStreetMap contributors. *Open Street Map*. 2017. URL: <https://www.openstreetmap.org>.

- [15] Markos Papageorgiou et al. «Review of road traffic control strategies». En: *Proceedings of the IEEE* 91.12 (2003), págs. 2043-2067.
- [16] PTV AG. *PTV Vissim 2022*. 2022. URL: <https://www.ptvgroup.com/es/soluciones/productos/ptv-vissim-nuevo/>.
- [17] QGIS Development Team. *QGIS Geographic Information System*. Open Source Geospatial Foundation, 2009. URL: <http://qgis.osgeo.org>.
- [18] Guido Van Rossum et al. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.
- [19] Fo Vo Webster. *Traffic signal settings*. Inf. téc. 1958.
- [20] Wolfram Research, Inc. *Mathematica 8.0*. Ver. 0.8. 2010. URL: <https://www.wolfram.com>.
- [21] Gregor Wunsch. «Coordination of traffic signals in networks». En: *Unpublished doctoral thesis, Technische Universität Berlin, Berlin* (2008).

# Apéndice A

## Código Python: Caso sobre una red

### Código Graficación

```
def diagram_e_t_o_i_ULTIMATE(artery): # ULTIMATE

    import matplotlib.pyplot as plt

    n=len(nodes_art[artery])

    nthetao=np.insert(thetao[artery], 0, 0)
    nthetai=np.insert(thetai[artery], 0, 0)

    # convierte distancia en un arreglo numpy con un cero al principio
    ndistancia=np.array(distancia[artery])
    ndistancia=np.insert(ndistancia, 0, 0)
    ndistancia=np.cumsum(ndistancia)

    # nuevo np.array de tiempos con un cero al inicio
    ntiemposo=np.insert(tiemposo[artery], 0, 0)
    ntiemposi=np.insert(tiemposi[artery], 0, 0)

    # Grafica las luces rojas y verdes (outbound)
    plt.figure(figsize=(12,8))
    for d in range(n):
        xr = np.array([0, redo[artery][d]])
        xg = np.array([redo[artery][d], 1])
        y = np.array([0, 0])
        for i in range(m):
            plt.plot(xr+0.5*(redo[artery][0]-redo[artery][d])+nthetao[d]+i,
                    ↪ y+ndistancia[d], color = 'red')
            plt.plot(xg+0.5*(redo[artery][0]-redo[artery][d])+nthetao[d]+i,
                    ↪ y+ndistancia[d], color = 'green')

    # Grafica las luces rojas y verdes (inbound)
    for d in range(n):
        xr = np.array([0, redi[artery][d]])-DELTAS[artery][0]
        xg = np.array([redi[artery][d], 1])-DELTAS[artery][0]
        y = np.array([0, 0])
        for i in range(m):
```

```

plt.plot(xr+0.5*(redi[artery][0]-redi[artery][d])+(1-nthetai[d])+i,
↪ y+ndistancia[d]-g, linestyle='dashed', color = 'red')
plt.plot(xg+0.5*(redi[artery][0]-redi[artery][d])+(1-nthetai[d])+i,
↪ y+ndistancia[d]-g, linestyle='dashed', color = 'green')

# Grafica el outbound bandwidth
x1=v+redo[artery][0]+value(model.wo[artery,nodes_art[artery][0]])
+ntiemposo[0]
y1=0
x2=x1+ntiemposo[0+1]
y2=ndistancia[0+1]
for d in range(n-1):
    xb = np.array([x1, x2])
    yb = np.array([y1, y2])
    plt.plot(xb, yb, color = 'blue')
    if d < n-2:
        x1=x2
        y1=y2
        x2=x1+ntiemposo[d+2]
        y2=ndistancia[d+2]

x1=v+redo[artery][0]+value(model.wo[artery,nodes_art[artery][0]])
+ntiemposo[0]+value(model.bo[artery])
y1=0
x2=x1+ntiemposo[0+1]
y2=ndistancia[0+1]
for d in range(n-1):
    xb = np.array([x1, x2])
    yb = np.array([y1, y2])
    plt.plot(xb, yb, color = 'blue')
    if d < n-2:
        x1=x2
        y1=y2
        x2=x1+ntiemposo[d+2]
        y2=ndistancia[d+2]

# Grafica el inbound bandwidth

x1=h-value(model.wi[artery,nodes_art[artery][0]])-ntiemposi[0] -
↪ DELTAS[artery][0]
y1=0
x2=x1-ntiemposi[0+1]
y2=ndistancia[0+1]
for d in range(n-1):
    xb = np.array([x1, x2])
    yb = np.array([y1, y2])
    plt.plot(xb, yb-g, color = 'blue')
    if d < n-2:
        x1=x2
        y1=y2
        x2=x1-ntiemposi[d+2]
        y2=ndistancia[d+2]

```

```

x1=h-value(model.wi[artery,nodes_art[artery][0]])
-ntiemposi[0]-value(model.bi[artery]) - DELTAS[artery][0]

y1=0
x2=x1-ntiemposi[0+1]
y2=ndistancia[0+1]
for d in range(n-1):
    xb = np.array([x1, x2])
    yb = np.array([y1, y2])
    plt.plot(xb, yb-g, color = 'blue')
    if d < n-2:
        x1=x2
        y1=y2
        x2=x1-ntiemposi[d+2]
        y2=ndistancia[d+2]

plt.xlabel("Tiempo (ciclos)")
plt.ylabel("Distancia de semáforos (metros)")
plt.show()

```

### Código Modelo Programación Lineal

```

T1=55
T2=75
df = pd.read_excel("parametroserb1.xlsx",sheet_name=None, header=None)
n=len(df["nodes_art"].index)
f1 =0.3
#Joints
J=df['J'].loc[0, :].values.tolist()
print(J)
df['edges_art'] = df['edges_art'].replace('{','(', regex=True)
df['edges_art'] = df['edges_art'].replace}','', regex=True)
for i in range(len(df['edges_art'].columns)):
    df['edges_art']=df['edges_art'].fillna(0)
    df['edges_art'][i]=df['edges_art'][i].map(lambda x: ast.literal_eval(x)
    if type(x) == str else x)

#forward
n2=len(df["forward"].index)
df['forward'] = df['forward'].replace('{','(', regex=True)
df['forward'] = df['forward'].replace}','', regex=True)
for i in range(len(df['forward'].columns)):
    df['forward']=df['forward'].fillna(0)
    df['forward'][i]=df['forward'][i].map(lambda x: ast.literal_eval(x)
    if type(x) == str else x)

#backward
n3=len(df["backward"].index)
df['backward'] = df['backward'].replace('{','(', regex=True)
df['backward'] = df['backward'].replace}','', regex=True)
for i in range(len(df['backward'].columns)):
    df['backward']=df['backward'].fillna(0)

```

```

df['backward'][i]=df['backward'][i].map(lambda x: ast.literal_eval(x)
if type(x) == str else x)

nodes_art={}
edges_art={}
forward={}
backward={}
distancia={}
redo={}
redi={}
lo={}
li={}
eo={}
ei={}
fo={}
fi={}
go={}
gi={}
ho={}
hi={}
for i in range(n):
    nodes_art[i+1]=list(y for y in
        ↪ np.array(df["nodes_art"].loc[i,].fillna(0),dtype="int")
    if y!=0)
    edges_art[i+1]=list(y for y in np.array(df['edges_art'].loc[i,].fillna(0))
    if y!=0)
    distancia[i+1]=list(y for y in
        ↪ np.array(df["distancias"].loc[i,].fillna(0),dtype="int") if y!=0)
    redo[i+1]=list(y for y in np.array(df["redo"].loc[i,].fillna(0),
        ↪ dtype='float') if y!=0)
    redi[i+1]=list(y for y in np.array(df["redi"].loc[i,].fillna(0),
        ↪ dtype='float') if y!=0)
    lo[i+1]=list(fl*y for y in np.array(df["lo"].loc[i,].fillna(0),
        ↪ dtype='float') if y!=0)
    li[i+1]=list(fl*y for y in np.array(df["li"].loc[i,].fillna(0),
        ↪ dtype='float') if y!=0)
    eo[i+1]=list(y for y in np.array(df["eo"].loc[i,].fillna(0), dtype='float')
        ↪ if y!=0)
    ei[i+1]=list(y for y in np.array(df["ei"].loc[i,].fillna(0), dtype='float')
        ↪ if y!=0)
    fo[i+1]=list(y for y in np.array(df["fo"].loc[i,].fillna(0), dtype='float')
        ↪ if y!=0)
    fi[i+1]=list(y for y in np.array(df["fi"].loc[i,].fillna(0), dtype='float')
        ↪ if y!=0)
    go[i+1]=list(y for y in np.array(df["go"].loc[i,].fillna(0), dtype='float')
        ↪ if y!=0)
    gi[i+1]=list(y for y in np.array(df["gi"].loc[i,].fillna(0), dtype='float')
        ↪ if y!=0)
    ho[i+1]=list(y for y in np.array(df["ho"].loc[i,].fillna(0), dtype='float')
        ↪ if y!=0)
    hi[i+1]=list(y for y in np.array(df["hi"].loc[i,].fillna(0), dtype='float')
        ↪ if y!=0)

```

```

print("nodes_art= ",nodes_art)
print("edges_art= ",edges_art)
print("redo= ",redo)
print("redi= ",redi)
print("lo= ",lo)
print("li= ",li)
print("eo= ",eo)
print("ei= ",ei)
print("fo= ",fo)
print("fi= ",fi)
print("go= ",go)
print("gi= ",gi)
print("ho= ",ho)
print("hi= ",hi)

for i in range(n2):
    forward[i+1]=list(y for y in np.array(df["forward"].loc[i,].fillna(0))
        if y!=0)
print("forward= ",forward)
for i in range(n3):
    backward[i+1]=list(y for y in np.array(df["backward"].loc[i,].fillna(0))
        if y!=0)
print("backward= ",backward)

art=range(1,n+1)

start =time.time_ns()

model=ConcreteModel()

model.nC=RangeSet(len(forward))
# *****Conjuntos

indexvar=[]
for i in art:
    for j in range(len(nodes_art[i])):
        indexvar.append((i,nodes_art[i][j]))
print("indexvar: ",indexvar)

model.ivs=Set(initialize = indexvar)

indexvar=[]
for i in art:
    for j in range(len(nodes_art[i])-1):
        indexvar.append((i,nodes_art[i][j]))
print("indexvar1: ",indexvar)

model.ivs1=Set(initialize = indexvar)

```

```

model.art=RangeSet(n)

model.R = Set(range(1,n+1))

for i in range(1,n+1):
    model.R[i] = nodes_art[i]

model.R1 = Set(range(1,n+1))
for i in range(1,n+1):
    nai,nai=nodes_art[i],nodes_art[i] [ : -1]
    model.R1[i] = nai

model.R2 = Set(range(1,n+1))
for i in range(1,n+1):
    nai,nai=nodes_art[i],nodes_art[i] [ : -2]
    model.R2[i] = nai

# *****Creamos las variables

model.bo = Var(model.art,within=NonNegativeReals)
model.bi = Var(model.art,within=NonNegativeReals)
model.wo = Var(model.ivs,within=NonNegativeReals)
model.wi = Var(model.ivs,within=NonNegativeReals)
model.z = Var(within=NonNegativeReals) #cycles/sec

model.to = Var(model.ivs1,within=NonNegativeReals)
model.ti = Var(model.ivs1,within=NonNegativeReals)

model.deltao = Var(model.ivs,within=Binary)
model.deltai = Var(model.ivs,within=Binary)
model.m = Var(model.ivs1,within=Integers)
model.C = Var(model.nC,within=Integers)
#Función Objetivo
def fo_rule(model):
    return sum(model.bo[i] + model.bi[i] for i in model.art)

model.fo=Objective(rule=fo_rule,sense=maximize)

def r1_rule(model,a):
    return model.bi[a] == 1*model.bo[a]
model.r1=Constraint(model.art,rule=r1_rule)

def rTU_rule(model):
    return model.z <= 1/T1
model.rTU=Constraint(rule=rTU_rule)
def rTL_rule(model):
    return model.z >= 1/T2
model.rTL=Constraint(rule=rTL_rule)

model.r= ConstraintList()
for a in model.art:

```

```

for i in model.R[a]:
    model.r.add(model.wo[a,i] + model.bo[a] <=
        ↪ 1-redo[a][nodes_art[a].index(i)])

for a in model.art:
    for i in model.R[a]:
        model.r.add(model.wi[a,i] + model.bi[a]
            ↪ <=1-redi[a][nodes_art[a].index(i)])

for a in model.art:
    for i in model.R1[a]:
        phis1s2o =
            ↪ model.wo[a,i]-model.wo[a,nodes_art[a][nodes_art[a].index(i)+1]]
        + model.to[a,i] + 0.5*redo[a][nodes_art[a].index(i)]
        - 0.5*redo[a][nodes_art[a].index(i)+1] #phio[i,i+1]
        phis1s2i =
            ↪ model.wi[a,i]-model.wi[a,nodes_art[a][nodes_art[a].index(i)+1]]
        + model.ti[a,i] + 0.5*redi[a][nodes_art[a].index(i)]
        - 0.5*redi[a][nodes_art[a].index(i)+1] #phii[i,i+1]
        DELTAs1 = 0.5*((2*model.deltao[a,i]-1)*lo[a][nodes_art[a].index(i)]
            - (2*model.deltai[a,i]-1)*li[a][nodes_art[a].index(i)]) #DELTA[i]
        DELTAs2 = 0.5*((2*model.deltao[a,nodes_art[a][nodes_art[a].index(i)+1]]-1)
            *lo[a][nodes_art[a].index(i)+1]
            - (2*model.deltai[a,nodes_art[a][nodes_art[a].index(i)+1]]-1)
            *li[a][nodes_art[a].index(i)+1]) #DELTA[i+1]
        model.r.add(phis1s2o + phis1s2i + DELTAs1 - DELTAs2 == model.m[a,i])

# Límites de velocidad
for a in model.art:
    for i in model.R1[a]:
        model.r.add(model.to[a,i] <= (distancia[a][nodes_art[a].index(i)]
            /eo[a][nodes_art[a].index(i)])*model.z)

for a in model.art:
    for i in model.R1[a]:
        model.r.add(model.to[a,i] >= (distancia[a][nodes_art[a].index(i)]
            /fo[a][nodes_art[a].index(i)])*model.z)

for a in model.art:
    for i in model.R1[a]:
        model.r.add(model.ti[a,i] <= (distancia[a][nodes_art[a].index(i)]
            /ei[a][nodes_art[a].index(i)])*model.z)

for a in model.art:
    for i in model.R1[a]:
        model.r.add(model.ti[a,i] >=(distancia[a][nodes_art[a].index(i)]
            /fi[a][nodes_art[a].index(i)])*model.z)

```

```

# Cambios de velocidad
for a in model.art:
    for i in model.R2[a]:
        model.r.add((distancia[a][nodes_art[a].index(i)]/distancia[a] \
                    [nodes_art[a].index(i)+1])* \
                    model.to[a,nodes_art[a][nodes_art[a].index(i)+1]]
                    - model.to[a,i] <= (distancia[a][nodes_art[a].index(i)]
                    /go[a][nodes_art[a].index(i)])
                    *model.z)

for a in model.art:
    for i in model.R2[a]:
        model.r.add((distancia[a][nodes_art[a].index(i)]/distancia[a] \
                    [nodes_art[a].index(i)+1])* \
                    model.to[a,nodes_art[a][nodes_art[a].index(i)+1]]
                    - model.to[a,i] \
                    >=(distancia[a][nodes_art[a].index(i)]
                    /ho[a][nodes_art[a].index(i)])*model.z)

for a in model.art:
    for i in model.R2[a]:
        model.r.add((distancia[a][nodes_art[a].index(i)]/distancia[a] \
                    [nodes_art[a].index(i)+1])* \
                    model.ti[a,nodes_art[a][nodes_art[a].index(i)+1]]
                    - model.ti[a,i] <= (distancia[a][nodes_art[a].index(i)]
                    /gi[a][nodes_art[a].index(i)])*model.z)

for a in model.art:
    for i in model.R2[a]:
        model.r.add((distancia[a][nodes_art[a].index(i)]/distancia[a] \
                    [nodes_art[a].index(i)+1])* \
                    model.ti[a,nodes_art[a][nodes_art[a].index(i)+1]]
                    - model.ti[a,i] \>=(distancia[a][nodes_art[a].index(i)]
                    /hi[a][nodes_art[a].index(i)])*model.z)

# ***** Cycle constraints
thetaF={}
thetaB={}
for c in model.nC:
    for e in range(len(forward[c])):
        a=[i for i in edges_art for j in edges_art[i] if j==forward[c][e]][0]
        print("F= ",a)
        thetaF[c,a,forward[c][e][0],forward[c][e][1]]=\
        (0.5*redo[a][nodes_art[a].index(forward[c][e][0])]+\
        model.wo[a,forward[c][e][0]]+\
        model.to[a,forward[c][e][0]]-\
        model.wo[a,forward[c][e][1]]-\
        0.5*redi[a][nodes_art[a].index(forward[c][e][1])])
    for e in range(len(backward[c])):
        a=[i for i in edges_art for j in edges_art[i] if j==backward[c][e]][0]

```

```

print("F= ",a)
thetaB[c,a,backward[c][e][0],backward[c][e][1]]=\
(0.5*redo[a][nodes_art[a].index(backward[c][e][0])]+\
model.wo[a,backward[c][e][0]]+\
model.to[a,backward[c][e][0]]-\
model.wo[a,backward[c][e][1]]-\
0.5*redi[a][nodes_art[a].index(backward[c][e][1])])
print(thetaF)
print(thetaB)

ft=list(thetaF.keys())
bt=list(thetaB.keys())
fb=ft+bt

junctions={} #(cycle, junction):(artery_m,artery_c)
for c in model.nC:
    for j in J:
        art=[]
        for k in range(len(fb)):
            if fb[k][0]==c:
                if fb[k][2]==j or fb[k][3]==j:
                    art.append(fb[k][1])
#                    print(art)
                junctions[c,j]=art
print("Junctions= ",junctions)

phif=sum(thetaF[i] for i in thetaF if i[0]==c)

DSPSI={} #Dictionary per cycle of sum(PSI)
for c in model.nC:
    SPSI=0
    for i in junctions:
        if i[0]==c:
            j=i[1]
            a=junctions[i][0]
            b=junctions[i][1]
            SPSI=SPSI+0.5-0.5*((2*model.deltai[b,j]-1)*
            li[b][nodes_art[b].index(j)]-
            (2*model.deltai[a,j]-1)*li[a][nodes_art[a].index(j)])
    DSPSI[c]=SPSI
print("DSPSI=*****",DSPSI)

for c in model.nC:
    phif=sum(thetaF[i] for i in thetaF if i[0]==c)
    phib=sum(thetaB[i] for i in thetaB if i[0]==c)
    model.r.add(phif - phib + DSPSI[c] == model.C[c])

results = SolverFactory('cplex_direct').solve(model)

end = time.time_ns()

print(f"Time taken is {end - start}")

```

```

print(nodes_art)
#####Showing results

print('FO= ',value(model.fo))
for a in model.art:
    print('bo[' ,a, ']= ',value(model.bo[a]))
    print('bi[' ,a, ']= ',value(model.bi[a]))
print('z= ',value(model.z))
print("T= ",1/value(model.z))

for a in model.art:
    for i in model.R1[a]:
        print('m[' ,a, ", ",i, ']', '=',round(abs(value(model.m[a,i])))

for a in model.art:
    for i in model.R[a]:
        print('wo[' ,a, ", ",i, ']', '=',round(abs(value(model.wo[a,i])),5))
for a in model.art:
    for i in model.R[a]:
        print('wi[' ,a, ", ",i, ']', '=',round(abs(value(model.wi[a,i])),5))

for a in model.art:
    for i in model.R1[a]:
        print('vo[' ,a, ', ',i, ', ',nodes_art[a][nodes_art[a].index(i)+1], ']=',
,round(abs(distancia[a][nodes_art[a].index(i)]*value(model.z)
/value(model.to[a,i])),5))
for a in model.art:
    for i in model.R1[a]:
        print('vi[' ,a, ', ',i, ', ',nodes_art[a][nodes_art[a].index(i)+1], ']=',
,round(abs(distancia[a][nodes_art[a].index(i)]*value(model.z)
/value(model.ti[a,i])),5))

for a in model.art:
    for i in model.R1[a]:
        print('to[' ,a, ', ',i, ', ',nodes_art[a][nodes_art[a].index(i)+1], ']=',
,round(abs(value(model.to[a,i])),5))
for a in model.art:
    for i in model.R1[a]:
        print('ti[' ,a, ', ',i, ', ',nodes_art[a][nodes_art[a].index(i)+1], ']=',
,round(abs(value(model.ti[a,i])),5))

for a in model.art:
    for i in model.R[a]:
        print('deltao[' ,a, ", ",i, ']= ',int(value(model.deltao[a,i])))
for a in model.art:
    for i in model.R[a]:
        print('deltai[' ,a, ", ",i, ']= ',int(value(model.deltai[a,i])))

# convierte los phis en theta(1,i)

thetao={}
for a in model.art:

```

```

thetaolist=np.array([])
po=0
for i in model.R1[a]:
    po=(po+0.5*redo[a][nodes_art[a].index(i)]+
        value(model.wo[a,nodes_art[a][nodes_art[a].index(i)]])+
        value(model.to[a,i])-
        value(model.wo[a,nodes_art[a][nodes_art[a].index(i)+1]])-
        -0.5*redo[a][nodes_art[a].index(i)+1])%1
    thetaolist=np.append(thetaolist,po)
thetao[a]=thetaolist
print("thetao= ",thetao)

thetai={}
for a in model.art:
    thetailist=np.array([])
    pi=0
    for i in model.R1[a]:
        pi=(pi+0.5*redi[a][nodes_art[a].index(i)]+
            value(model.wi[a,nodes_art[a][nodes_art[a].index(i)]])+
            value(model.ti[a,i])-
            value(model.wi[a,nodes_art[a][nodes_art[a].index(i)+1]])-
            0.5*redi[a][nodes_art[a].index(i)+1])%1
        thetailist=np.append(thetailist,pi)
    thetai[a]=thetailist
print("thetai= ",thetai)

# Convierte la salida de tiempos to y ti en listas

tiemposo={}
for a in model.art:
    tiemposolist=np.array([])
    for i in model.R1[a]:
        tiemposolist=np.append(tiemposolist,value(model.to[a,i]))
    tiemposo[a]=tiemposolist
print("tiemposo= ",tiemposo)

tiemposi={}
for a in model.art:
    tiemposilist=np.array([])
    for i in model.R1[a]:
        tiemposilist=np.append(tiemposilist,value(model.ti[a,i]))
    tiemposi[a]=tiemposilist
print("tiemposi= ",tiemposi)

# Convierte la salida de DELTAS en una lista
DELTAS={}
for a in model.art:
    DELTASlist=np.array([])
    for i in model.R[a]:
        DELTASlist =np.append(DELTASlist,0.5*((2*value(model.deltao[a,i])-1)
            *lo[a][nodes_art[a].index(i)]-(2*value(model.deltai[a,i])-1)*
            li[a][nodes_art[a].index(i)))) #DELTA[i]
    DELTAS[a]=DELTASlist

```

```
print("DELTAS= ",DELTAS)

# Imprime el plot espacio-tiempo
m=5 # número de periodos graficados
v=0 # número de periodos adelante para graficar el outbound desde el primer
    ↪ periodo {0,1,...}
h=4 # número de periodos adelante para graficar el inbound {1,2,...}
g=30 # distancia para graficar las señales de outbound e inbound

for i in range(1,n+1):
    diagram_e_t_o_i_ULTIMATE(i)
```

## Apéndice B

# Código Grafo y Base de Ciclos Wolfram Mathematica

```
g=Graph[{1->2,2->3,2->4,4->5,5->6,6->7,7->8,9->8,1->9,5->9},
VertexCoordinates->{{1,1},{2,0.7},{2.5,0.4},{2.5,2},{2.5,3},
{2.8,4},{3,5},{1.3,6},{1.3,3.3}},VertexLabels->Placed[Automatic,Center]
,EdgeLabelStyle -> Directive[12, Background -> White]
,VertexSize-> .20,VertexStyle->White,EdgeStyle->Black]
```

```
p1=Flatten[{Pick[EdgeList[g],#,1]} & /@ -EdgeCycleMatrix[g],1]
p2=Flatten[{Pick[EdgeList[g],#,-1]} & /@ -EdgeCycleMatrix[g],1]
a=HighlightGraph[g,Style[#,Darker[Blue],Thick]] & /@p1
b=HighlightGraph[g,Style[#,Darker[Red],Thick]] & /@p2
For[i=1,i<=Length[p2],i++,Print[HighlightGraph[a[[i]],Style[p2[[i]]
,Darker[Red],Thick]]]]
F=p1/.DirectedEdge[x_,y_]->{x,y}
B=p2/.DirectedEdge[x_,y_]->{x,y}
M=Normal[-EdgeCycleMatrix[g]]
c=Join[{M[[1]]},{M[[2]]}]
J={1,2,3,4,5,6,7,8,9}
Export["cyclebasis.xls", {"cycles" -> c, "forward" -> F,
"backward"->B,"J"->J}]
CopyFile["cyclebasis.xls",CloudObject["cyclebasis.xls"]
,OverwriteTarget->True]
```