

## Capítulo 3.

# Modelos de Sincronización Simultánea y con Intervalo de Tiempo.

### 3.1. Introducción.

Actualmente, la sincronización es la tarea más compleja dentro de la programación del tránsito, por lo que es bastante frecuente que se busque una dirección más intuitiva, sin llegar a ser una resolución exacta, tal y como se pretende en este documento. La programación, de hecho, intenta crear los tiempos de salida en las tablas de tiempos, mientras cumplimos con las siguientes premisas:

- Los requerimientos de frecuencia.
- La asignación eficiente de los viajes para cada uno de los autobuses.
- La sincronización de las llegadas seguras.

Aunque en este proyecto se profundice en los modelos desarrollados por Ceder (2001) y Eranki (2004), debemos tener en cuenta que existen otros autores que han barajado distintas opciones para optimizar el número de sincronizaciones; podemos hacer referencia al modelo desarrollado por Voss (1992), en el cual formula el problema de minimización de espera de tiempo de los pasajeros en los nodos de transferencia de la red, como un problema de asignación cuadrática; de forma similar a como fue expuesto por Lawer (1963), Hillier y Connors (1966). Su estudio hace referencia a los casos, donde cada autobús de la ruta,  $i$ , está unido por un conjunto de posibles tiempos de salida. Este problema fue modificado hacia un caso donde diferentes rutas, usaban parte de las huellas de la anterior, de tal forma que se tenían que establecer unos tiempos de seguridad.

Desilet y Rousseau (1992), describen un modelo diferente, el cual se establece a partir de una selección de tiempos de comienzos, que forman parte de un conjunto de

tiempos de comienzos posibles,  $T$ . La función objetivo se expresa de tal forma, que se minimicen las penalizaciones asociadas de la línea  $i$  a la línea  $j$ , para cada  $i, j$ . La función de penalización se puede calcular por varios caminos.

Daganzo (1990) presenta el problema de coordinación de una red que comprende un solo nodo, de tal forma que existen rutas entrantes y salientes. Partiendo de este modelo, Lee y Schonfeld (1991) intentan la sincronización de un autobús de una ruta, con la línea de tren, mientras suponen una serie de condiciones estocásticas. Sus conclusiones establecían que no había una justificación para la sincronización, cuando nos encontrábamos con situaciones que se caracterizaban por un elevado número de variables de tiempos de llegada. Finalmente, Chin y Schonfeld (1998) ensayan la optimización del total de los costes, mientras integran el programa de una línea de tren, conjugada con los autobuses, demostrando la complejidad de este problema.

Estos autores no buscan la sincronización de las redes de transporte a través de un programa práctico; de tal forma que se ha orientado la búsqueda de la máxima sincronización a través de autores como Ceder y Eranki, cuyos modelos son la base de este proyecto.

Se presentarán dos modelos distintos, los cuales permitirán programas de tránsito que irán acompañados con un conjunto de restricciones ligadas a la cabecera de cada ruta; introducimos diferentes frecuencias para cada ruta, aplicando además otras limitaciones. La función objetivo será maximizar el número de llegadas simultáneas de los autobuses de las distintas rutas, para el global de la red. Este número de sincronizaciones variará dependiendo del modelo que estemos usando, tal y como expondremos a continuación; puesto que Ceder (2001) exige la sincronización simultánea, mientras Eranki (2004) permite un rango de tiempo permitido para cada nodo.

Una vez sentadas las bases de ambos modelos, pasamos a desarrollarlos de manera breve, ya que posteriormente en la documentación se establecerán las diferentes pautas a seguir con cada uno de ellos. En primer lugar, veremos cómo podemos calcular las frecuencias de paso de los vehículos.

- Mantener la calidad de servicio adecuada.
- Minimizar el número de vehículos necesarios.
- Se busca la agrupación de datos por franjas horarias.

Los datos de partida para ambos modelos serán prácticamente idénticos, ya que la diferencia radicará en los intervalos de tiempos establecidos para cada nodo, en el Modelo de sincronización para un intervalo de tiempo permitido.

- Identificar el número de paradas.
- Conocer la distancia entre paradas.
- Demanda de pasajeros por parada.
- Calidad del servicio. Medido para cada línea y franja horaria.

Con estas premisas se buscará encontrar las frecuencias de paso de vehículos para cada línea y franja horaria.

### 3.2. Modelo de Sincronización Simultánea.

Partiendo del modelo desarrollado por Ceder (2001), se partirá de una red de autobuses, la cual está presentada por un grafo,  $G = \{A, \tilde{N}\}$ , donde  $A$  es un conjunto de arcos representativos del viaje, de las rutas de los autobuses; mientras  $\tilde{N}$  es el conjunto de nodos de transferencia de la red.

Los datos con los que parte el problema serían los siguientes:

1.  $T$  siendo el horizonte de planificación. Los tiempos de salida de los autobuses pueden estar dentro de un intervalo discreto  $[0, T]$
2.  $M$  es el número de rutas de autobuses en la red.
3.  $N$  es el número de nodos de transferencia en la red.
4.  $Hmin_K$  se caracteriza como el mínimo avance entre dos salidas adyacentes de autobuses en una ruta  $K$  ( $1 \leq K \leq M$ ).
5.  $Hmax_K$  se caracteriza como el máximo avance permitido entre dos salidas adyacentes de autobuses en la ruta  $K$  ( $1 \leq K \leq M$ ).
6.  $F_K$  es el número de salidas que son programadas para la ruta  $K$  durante el intervalo  $[0, T]$  ( $1 \leq K \leq M$ ).
7.  $T_{kj}$  es el tiempo de viaje desde el punto de comienzo de la ruta  $K$  hasta el nodo  $j$  ( $1 \leq K \leq M, 1 \leq j \leq N$ ). Se considera determinista el tiempo de viaje.

Para poder desarrollar este modelo debemos asumir que la primera salida en cada ruta  $K$  debe estar dentro del intervalo  $[0, Hmax_k]$ .

Para cada  $K$ , nos encontramos con las siguientes restricciones:

$$Hmax_k \geq Hmin_k \quad (1)$$

$$T \geq (F_k - 1) * Hmin_k \quad (2)$$

$$T < F_k * Hmax_k \quad (3)$$

En el caso de la ruta  $K$  no pasase por el nodo  $j$ , se representaría de la siguiente manera  $T_{Kj} = -1$ .

Las variables de decisión serían las siguientes:

- $X_{ik}$  representa el tiempo de salida del autobús  $i$ -ésimo en la ruta  $K$  ( $1 \leq i \leq F_k$ ).
- $Z_{ikjqn}$  es una variable binaria que toma el valor 1 si el autobús  $i$ -ésimo de la ruta  $K$  se encuentra con el autobús  $j$ -ésimo de la ruta  $q$  en el nodo  $n$ ; tomando el valor 0 para cualquier otro caso.

Se Conseguirá  $A_{kq} = \{ n: 1 \leq n \leq N, T_{kn} \geq 0, T_{qn} \geq 0 \}$ .

Con estos datos se podrá realizar una formulación inicial del problema, según el modelo de sincronización simultánea.

$$\max \sum_{k=1}^{M-1} \sum_{i=1}^{F_k} \sum_{q=k+1}^M \sum_{j=1}^{F_q} \sum_{n \in A_{kn}} Z_{ikjqn}$$

s.a.

$$X_{1k} \leq Hmax_k, \quad 1 \leq k \leq M \quad (4)$$

$$X_{F_k n} \leq T, \quad 1 \leq k \leq M \quad (5)$$

$$Hmin_n \leq X_{(i+1)k} - X_{ik} \leq Hmax_k, \quad 1 \leq k \leq M, \quad 1 \leq i \leq F_k - 1 \quad (6)$$

$$Z_{ikjqn} = \max[1 - |(X_{ik} + T_{kn}) - (X_{jq} + T_{qn})|, 0]. \quad (7)$$

La restricción (4) nos asegura que el tiempo de salida no puede ir más allá del avance máximo del comienzo del horizonte de tiempo.

La restricción (5) nos asegura que la última salida está comprendida en el horizonte de planificación.

El contraste a estas dos restricciones se encuentra en la número (6), ya que nos indicará los límites de avance; con la restricción (7) se busca definir las variables binaria de la función objetivo.

Este modelo puede ser simplificado, si definimos una variable  $Y_{kq}$ , que representa el número total de llegadas simultáneas de autobuses de la ruta  $k$  con autobuses de la ruta  $q$ . El modelo cambiaría como sigue:

$$\text{Max } \sum_{k=1}^{M-1} \sum_{q=k+1}^M Y_{kq}$$

s.a.

$$Y_{kq} = \sum_{n \in A_{kq}} \sum_{i=1}^{F_k} \sum_{j=1}^{F_q} \max [1 - |(X_{ik} + T_{kn}) - (X_{jq} + T_{qn})|, 0] \quad (8)$$

Las restricciones (4) – (7) no sufren ningún cambio.

Esta última formulación NO LINEAL es la que desarrolla Ceder (2001). En este proyecto se plantea su linealización para poder ser resuelta mediante un paquete de programación lineal. Por tanto, para poder resolverlo, tenemos que reformularlo, haciendo uso de las siguientes variables binaria  $D_{ntjkq}$ , y una cota superior denotada por  $B$  ( $B = T + \max_{i,j} T_{ij}$ ). Con esto logramos, que la restricción (8) pase a ser de la siguiente manera:

$$B \cdot D_{ntjkq} \geq X_{ik} + T_{kn} - (X_{jq} + T_{qn}), \quad (9)$$

$$B \cdot D_{ntjkq} \geq X_{jq} + T_{qn} - (X_{ik} + T_{kn}), \quad (10)$$

$$Y_{kq} \leq \sum_{n \in A_{kq}} \sum_{i=1}^{F_k} \sum_{j=1}^{F_q} (1 - D_{ntjkq}) \quad (11)$$

Si  $X_{ik} + T_{kn} = X_{jq} + T_{qn}$ , tenemos la situación de llegada simultánea del autobús  $i$ -ésimo en la ruta  $k$  con el autobús  $j$ -ésimo de la ruta  $q$  en el nodo  $n$ . La variable  $D_{ntjkq}$  toma el valor 0, quedando por tanto, la variable  $Y_{kq}$  con el valor 1, acorde con la restricción (11).

Si  $X_{ik} + T_{kn} \neq X_{jq} + T_{qn}$ , por tanto las llegadas no coinciden, y la variable  $D_{nifkq}$  toma el valor que satisfaga las restricciones (9) y (10). El número de llegadas simultáneas entre autobuses de las rutas  $k$  y  $q$  ( $Y_{kq}$ ) no se incrementa en la restricción (11).

El hándicap de este modelo, se encuentra en que tiene un número muy extenso de variables binarias; de tal manera que la variable  $D_{nifkq}$  representa las llegadas simultáneas de los autobuses  $i$ -ésimos de la ruta  $k$  y los autobuses  $j$ -ésimos de la ruta  $q$  en el nodo  $n$ . Esto significa que se tendrá una variable entera para cada combinación de dos autobuses de diferentes rutas, que se interceptan en el nodo  $n$ , resultando un número de variables enteras muy grande. Este modelo resulta complejo a la hora de ejecutarlo, por lo que, a partir de cierto tamaño, se hará necesaria una heurística para poder resolverlo.

### 3.3. Modelo de sincronización con intervalo de tiempo dado.

Basándose en el modelo de Eranki (2004), se planteará un modelo que buscará la sincronización del transporte, ahora bien la coincidencia de las rutas se establece como un intervalo de tiempo, con esto se busca abrir un poco la holgura de la sincronización de las rutas en los distintos nodos.

Por tanto, sigue la línea establecida por Ceder (2001) pero sin especificar como sincronización, que los autobuses lleguen de manera exacta al nodo.

Tenemos la red de transporte (autobuses), que se representa por medio de un grafo  $G = \{A, \tilde{N}\}$ , donde  $A$  es un conjunto de arcos representativos del viaje, de las rutas de los autobuses; mientras  $\tilde{N}$  es el conjunto de nodos de transferencia de la red. Partiendo de este grafo, podemos desarrollar el modelo, con los siguientes datos de partida:

1.  $k = 1, \dots, M$ . Representando el número de rutas de la red.
2.  $q = 1, \dots, M$ . Representando el número de rutas de la red.
3.  $n = 1, \dots, N$ . Número de nodos de la red.
4.  $T$  siendo el horizonte de planificación. Los tiempos de salida de los distintos autobuses deben estar dentro de un intervalo discreto  $[0, T]$
5.  $Hmin_k$  se caracteriza como el mínimo avance entre dos salidas adyacentes de autobuses en una ruta  $K$  ( $1 \leq K \leq M$ ).
6.  $Hmax_k$  se caracteriza como el máximo avance permitido entre dos salidas adyacentes de autobuses en la ruta  $K$  ( $1 \leq K \leq M$ ).
7.  $F_k$  es el número de salidas que son programadas para la ruta  $K$  durante el intervalo  $[0, T]$  ( $1 \leq K \leq M$ ).
8.  $i, j = 1, \dots, F_k$ . Define las salidas de autobuses que se deben programar en el intervalo  $[0, T]$  de la ruta  $k$ .
9.  $T_{kn}$  siendo el tiempo de viaje desde la cabecera de la ruta  $k$  al nodo  $n$ .
10.  $A_{kq} = \{ n: 1 \leq n \leq N, T_{kn} \geq 0, T_{qn} \geq 0 \}$ . Estamos representando el conjunto de nodos comunes a las rutas  $k$  y  $q$ .
11.  $WTmin_n$ , se define como el intervalo de tiempo mínimo permitido de espera en el nodo  $n$ .
12.  $WTmax_n$ , se define como el intervalo máximo de tiempo permitido de espera en el nodo  $n$ .

Una vez que se tengan los datos del modelo, podemos pasar a definir las variables con las que se van a trabajar.

- $X_{tk}$  se expresa como el instante de salida del autobús i-ésimo de la ruta k  
(  $1 \leq i \leq F_k, 1 \leq k \leq M$ )
- $Z_{tkjqn} (= 1)$ . Es la variable de la función objetivo, de tal forma que se define como si el autobús i-ésimo de la ruta k y el autobús j-ésimo de la ruta q llegan al nodo n con una separación de tiempos admisibles (  $1 \leq i \leq F_k, 1 \leq k \leq M - 1, 1 \leq j \leq F_q, k + 1 \leq q \leq 1 n \in A_{kq}$ )

Con las variables definidas, se podrá desarrollar el modelo, como una función objetivo más una serie de restricciones.

$$\text{Max } \sum_{t=1}^{F_k} \sum_{k=1}^{M-1} \sum_{j=1}^{F_q} \sum_{q=k+1}^M \sum_{n \in A_{kq}} Z_{tkjqn}$$

s.a.

$$X_{1k} \leq Hmax_k, \quad k = 1, \dots, M \quad (1)$$

$$X_{F_k, k} \leq T, \quad k = 1, \dots, M \quad (2)$$

$$Hmin_k \leq X_{t+1, k} - X_{t, k} \leq Hmax_k, \quad k = 1, \dots, M; t = 1, \dots, F_k - 1$$

(3)

$$Z_{tkjqn} = \begin{cases} 1 & \text{si } WTmin_n \leq |(X_{tk} + T_{kn}) - (X_{jq} + T_{qn})| \leq WTmax_n \\ 0 & \text{si } WTmin_n > |(X_{tk} + T_{kn}) - (X_{jq} + T_{qn})| \text{ OR } WTmax_n < |(X_{tk} + T_{kn}) - (X_{jq} + T_{qn})| \end{cases}$$

(4)

Como se podrá observar, el modelo desarrollado por Eranki (2004) no es lineal, por lo tanto para poder trabajar con él, y extraer los resultados necesitamos linealizar dicho modelo; ya que se partirá de un modelo de maximización de sincronizaciones, que contiene un conjunto de restricciones que no son lineales, y por lo tanto, no serán de utilidad para lograr una solución óptima. Para ello, en este documento se linealiza el modelo anterior a través de unas variables continuas positivas  $u$  y  $v$ , que recogerán la diferencia de tiempos entre las llegadas de cada dos autobuses a cada nodo.

$$(X_{ik} + T_{kn}) - (X_{jq} + T_{qn}) = u_{ntjka} - v_{ntjka}; u_{ntjka} \geq 0; v_{ntjka} \geq 0$$

$$|(X_{ik} + T_{kn}) - (X_{jq} + T_{qn})| = u_{ntjka} + v_{ntjka}$$

$$\begin{cases} \text{si } u_{ntjka} > 0 \Rightarrow v_{ntjka} = 0 \\ \text{si } v_{ntjka} > 0 \Rightarrow u_{ntjka} = 0 \end{cases} \Leftrightarrow \begin{cases} 0 \leq u_{ntjka} \leq K\gamma_{ntjka} \\ 0 \leq v_{ntjka} \leq K(1 - \gamma_{ntjka}) \\ \gamma_{ntjka} = 0, 1; K \text{ cota superior} \end{cases}$$

Las variables binarias de este modelo son:

- $\gamma_{kq}$  Siendo el número total de coincidencias de autobuses de la ruta k y de la ruta q, dentro de los márgenes permitidos en nodos comunes.
- $D_{ntjka}$  ( $= 1$ ). Si el autobús i de la ruta k coincide dentro de los márgenes con el autobús j de la ruta q, en el nodo n.
- $\gamma_{ntjka} = 1$ . Si el autobús i de la ruta k llega después al nodo n que el autobús j de la ruta q.

La linealización de las restricciones (4) se realiza de la siguiente forma:

$$(4') \quad u_{ntjfkq} + v_{ntjfkq} \leq WTmin_n, \quad \text{OR}$$

$$\begin{aligned} & u_{ntjfkq} + v_{ntjfkq} > WTmax_n \Rightarrow D_{ntjfkq} = 1 \Leftrightarrow st D_{ntjfkq} = 0 \Rightarrow \\ & \begin{cases} WTmin_n \leq u_{ntjfkq} + v_{ntjfkq} \\ WTmax_n \geq u_{ntjfkq} + v_{ntjfkq} \end{cases} \\ & \Leftrightarrow \begin{cases} WTmin_n - u_{ntjfkq} - v_{ntjfkq} \leq B \cdot D_{ntjfkq} \\ -WTmax_n + u_{ntjfkq} + v_{ntjfkq} \leq B \cdot D_{ntjfkq} \end{cases} \text{ siendo } B = T + \max_{k,n} \{T_{kn}\} \text{ una cota} \end{aligned}$$

superior.

La función objetivo quedaría de la siguiente manera:

$$\text{Max } \sum_{k=1}^{M-1} \sum_{q=k+1}^M Y_{kq}$$

s.a

$$Y_{kq} \leq \sum_{i=1}^{F_k} \sum_{j=1}^{F_q} \sum_{ns \in A_{kq}} (1 - D_{ntjfkq})$$

Con esto podemos desarrollar el modelo lineal, tal y como sigue:

$$\text{Max } \sum_{k=1}^{M-1} \sum_{q=k+1}^M Y_{kq}$$

s.a

$$X_{1k} \leq Hmax_k, \quad k = 1, \dots, M \quad (1)$$

$$X_{F_k, k} \leq T, \quad k = 1, \dots, M \quad (2)$$

$$Hmin_k \leq X_{t+1, k} - X_{t, k} \leq Hmax_k, k = 1, \dots, M; t = 1, \dots, F_k - 1 \quad (3)$$

$$(X_{tk} | T_{kn}) \quad (X_{jq} | T_{qn}) = u_{ntjfkq} \quad v_{ntjfkq}; k = 1, \dots, M; t = 1, \dots, F_k; q = k + 1, \dots, M; j = 1, \dots, F_q; ns \in A_{kq}$$

(4)

$$B \cdot D_{ntjfkq} \geq -u_{ntjfkq} - v_{ntjfkq} + WTmin_n \quad (5)$$

$$B \cdot D_{ntjfkq} \geq u_{ntjfkq} + v_{ntjfkq} - WTmax_n \quad (6)$$

$$\begin{cases} 0 \leq u_{ntjfkq} \leq K Y_{ntjfkq} \\ 0 \leq v_{ntjfkq} \leq K(1 - Y_{ntjfkq}) \\ Y_{ntjfkq} = 0, 1; K \text{ cota superior} \end{cases} \quad (7) \text{ y } (8)$$

$$Y_{kq} \leq \sum_{i=1}^{F_k} \sum_{j=1}^{F_q} \sum_{n \in A_{kq}} (1 - D_{nt/jkq}); \quad k = 1, \dots, M - 1; q = k + 1, \dots, M \quad (9)$$

Donde  $D_{nt/jkq} = (0,1); X_{tk} \geq 0;$

$Y_{kq} \geq 0; u_{nt/jkq} \geq 0; v_{nt/jkq} \geq 0; \text{ enteros}; \gamma_{nt/jkq} = (0,1)$

### 3.4.- Algoritmo aproximado para la sincronización.

Los modelos expuestos anteriormente resuelven el problema de forma óptima cuando se considera que las llegadas de dos autobuses a un nodo tienen que coincidir (Ceder) o existe una holgura entre llegadas de autobuses (Eranksi).

El número excesivo de variables binarias recomienda el uso de un algoritmo aproximado de resolución de estos problemas. Para ello, se ha desarrollado en lenguaje C un algoritmo basado en Eranksi (2004) que contempla ambos problemas. En la siguiente figura se muestra el diagrama de flujo del algoritmo:

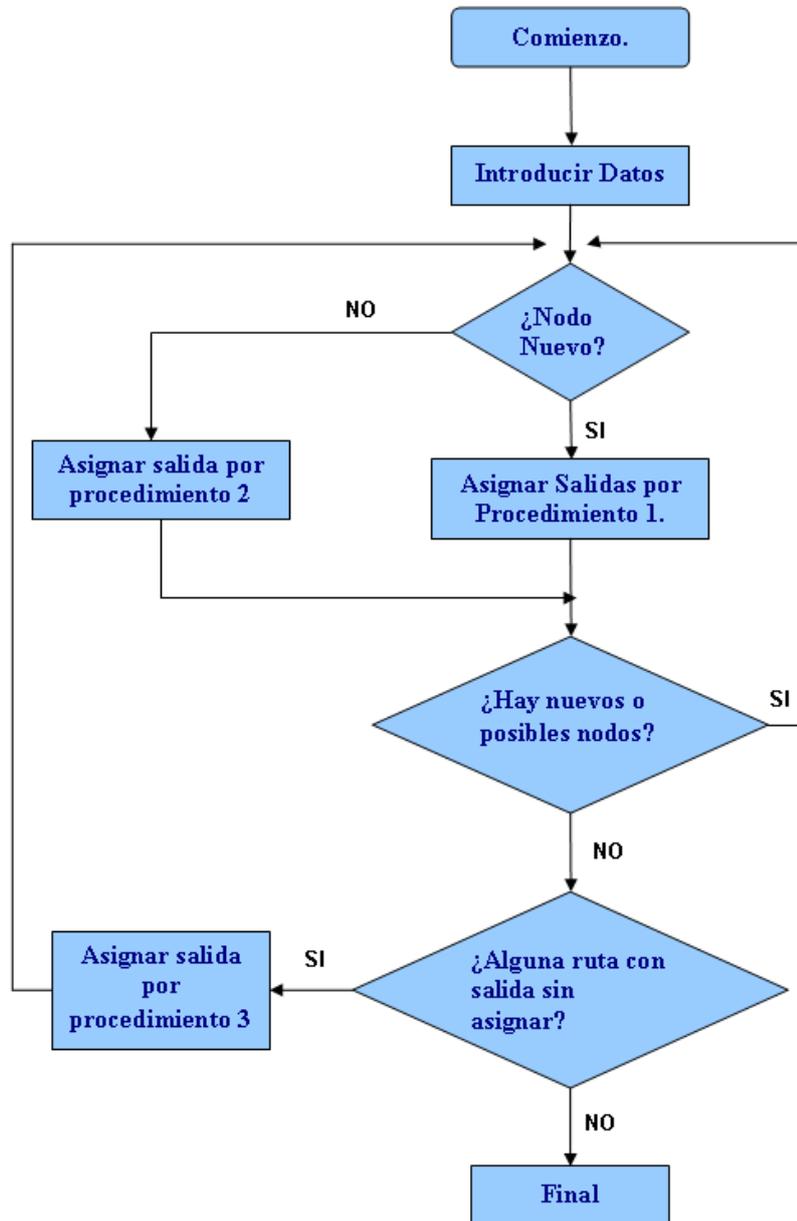


Figura 3.1. Diagrama de Flujo del Algoritmo aproximado de Eranki

Se va a proceder a continuación a explicar cada uno de los módulos del algoritmo, expresados en el anterior diagrama de flujo.

Las restricciones que debe cumplir el programa serán las siguientes:

1.  $Hmax_i \geq Hmin_i$
2.  $T \geq (f_i - 1) * Hmin_i$
3. El límite máximo posible para el horizonte de planificación, es el máximo valor dado por  $(f_i * Hmax_i)$  entre todas las rutas.
4. La espera de tiempo mínima posible para cada nodo, será siempre un valor mayor que cero.
5. La máxima espera de tiempo posible para cada nodo, es el valor que no excede el mayor avance de las rutas que pasan a través de dicho nodo.

### 3.4.1. Procedimiento de Selección de un Nodo.

En cada iteración del algoritmo, se seleccionará un nodo a través de todos los nuevos y posibles nodos; de tal forma que existen tres pasos para seleccionarlos. Cada uno de estos pasos se puede desarrollar de la siguiente forma:

#### PASO 1.

Entre los “nuevos” y “posibles” nodos, buscaremos aquellos que tienen el máximo número de llegadas de tiempo ya establecidas. Si no tuviéramos establecidas las llegadas de tiempo, o existieran vínculos, pasaríamos directamente al “paso 2”.

Ahora bien, si en esta etapa se ha seleccionado un “nuevo” o “posible” nodo, se le etiqueta como “Nodo Seleccionado ( $k^*$ )”, y el programa se sale de las etapas “paso”, y busca de forma directa los distintos “procedimientos”.

La finalidad de este “paso 1” consiste en buscar si tenemos algún nodo seleccionado, para poder comenzar el proceso iterativo con el primer procedimiento.

#### PASO 2

En el “paso 2” identificamos el nodo con el número máximo de rutas que están pasando a través de él. Si existiera algún vínculo, nos iríamos al “paso 3”.

Sólo si uno de los nodos ha sido identificado, se le etiqueta como “Nodo Seleccionado ( $k^*$ )”, e inmediatamente el programa sale de esta parte del código, y va directamente al “procedimiento”.

En este caso, al ejecutar el paso anterior y no tener un nodo seleccionado, lo que hace es buscar el primer nodo no seleccionado, por el cual pasen mayor número de rutas. Si existe un único nodo en estas circunstancias, pasamos directamente a la iteración del programa; ahora bien, si por cualquier causa tuviéramos un empate, pasaríamos directamente al “paso 3”, para poder escoger un nodo dentro de los ya seleccionados del empate.

Se considerará que existe un empate, cuando tenemos más de un nodo, por el cual pasan el mismo número de rutas, resultando ser además el número de rutas máximas que pasan por cualquiera de los nodos chequeados.

### PASO 3

El “paso 3” es el último paso para poder seleccionar un nodo; para ello, se calcula el máximo tiempo de viaje desde el origen de cada una de las rutas hasta los distintos nodos, por los que pasan. La selección se realizará sobre el nodo que tenga el mínimo valor, etiquetándolo por tanto, como “Nodo Seleccionado ( $k^*$ )”.

Si existiera cualquier tipo de vínculo se saldría arbitrariamente.

Una vez que tenemos un nodo seleccionado, el programa comienza cada uno de los procedimientos, para poder obtener la sincronización de las rutas de transporte, los tiempos de salida, y los tiempos entre cada uno de los autobuses.

Con este paso lo que hacemos es deshacer el empate, si lo hubiera, en el número de nodos candidato a ser seleccionado.

En este paso se utiliza la variable  $tin\_max[j]$ , que viene dada de un bucle *for* anterior al proceso iterativo de *While*. Esta variable nos devuelve el tiempo máximo de viaje para cada autobús de las distintas rutas, que pasan por los diferentes nodos; proporcionándonos también la ruta con ese tiempo máximo ( $ruta\_tin\_max[j]$ ).

Con cualquiera de estos pasos llegamos a obtener un nodo seleccionado, estando en ese momento en las condiciones adecuadas para poder realizar los distintos procedimientos, dependiendo de las distintas condiciones.

### 3.4.2. Procedimientos de resolución.

#### Procedimiento 1.

En este procedimiento se busca, asignar los tiempos de salida para las rutas que pasan por el nodo seleccionado, si este es nuevo. Se supondrá que tenemos dos rutas que pasan por dicho nodo, luego este procedimiento lo que hace es asignar el tiempo de salida a la ruta que toma el máximo tiempo de llegada al nodo. Se impondrá, además que el tiempo de salida para el primer autobús de dicha la ruta sea cero, tomando por tanto, el máximo tiempo de viaje.

Para la otra ruta se especifica, que el tiempo de salida asignado sea tal que los tiempos de llega de ambas rutas al nodo seleccionado, esté siempre entre los valores mínimo y máximo permitidos.

Lo que se busca es realizar un primer chequeo, para identificar los tiempos mínimos de espera, permitidos, y con ello, si es posible, asignar los tiempos de salida de los autobuses de forma acorde al margen de tiempo establecido por el nodo seleccionado.

En primer lugar, el procedimiento busca chequear para encontrar, si es posible, un tiempo mínimo de espera permitido; si esto es posible, se asigna el tiempo de salida de forma acorde. Si esto no se satisface se incrementa el tiempos para uno (tiempo discreto), y verificamos para el siguiente, la posible espera de tiempo desde el límite, y hasta que el máximo sea alcanzado. Las sub secuencias de salidas para esas rutas estarán mezcladas después, un tiempo  $d$  desde la última salida. Por tanto, lo que busca este “procedimiento” es el mínimo posible de, que viene dado por:

$$d = \min_{i=1,2,\dots,M} [\max_{i=1,2,\dots,M} (H_{\min,i}), \min_{i=1,2,\dots,M} (H_{\max,i})] \quad \forall i, \text{ pasando a través del nodo seleccionado.}$$

Este procedimiento efectúa los siguientes pasos:

Paso 1: Nodo seleccionado ( $k^*$ ).

Para todas las rutas  $i$ , que pasan por el nodo seleccionado, se calculará el mínimo valor posible de  $d$ , que satisfaga:

$$d = \min_{i=1,2,\dots,M} [\max_{i=1,2,\dots,M} (H_{\min,i}), \min_{i=1,2,\dots,M} (H_{\max,i})] \quad \forall i, \text{ pasando a través del nodo seleccionado.}$$

Establecer el máximo tiempo: el máximo tiempo de viaje alcanzado en  $k^*$ , identificando además a la ruta que va asociada a este máximo tiempo alcanzado, por medio de la etiqueta  $i^*$

Paso 2: Para la ruta  $i^*$ , se establece la primera salida ( $p=1$ ) como  $X_{i^*1} = 0$ .

Paso 3: Para el resto de rutas  $i$ , que pasan a través de dicho nodo  $k^*$ , tenemos lo siguiente:

Si  $(\text{maxtime} - \text{WT } \min_{k^*} - t_{ik^*} > 0)$

Alcanzamos  $X_{i1} = (\text{maxtime} - \text{WT } \min_{k^*} - t_{ik^*} > 0)$  y saltamos hacia el paso 5.

De otra manera, alcanzamos  $w = \text{WT } \min_{k^*}$

Paso 4: Si  $(\text{maxtime} + w) \geq t_{ik}^*$  y si  $(\text{maxtime} + w - t_{ik}^*) \leq H_{\text{max}_i}$

Alcanzamos  $X_{i1} = (\text{maxtime} + w - t_{ik}^*)$  y nos movemos hacia el paso 5.

Si no, alcanzamos  $w = w + 1$ , si  $w \leq WT_{\text{max}_k^*}$ , repetimos el paso 4, y si no salimos.

Paso 5: Para aquellas rutas  $i$  e  $i^*$ , si el procedimiento ha podido encontrar el valor de  $d$  en el paso 1, la sub secuencia de salidas será, por ejemplo  $\forall p=2, \dots, \min(f_i, f_{i^*})$ , asignándose después un intervalo de valor  $d$  desde la salida previa. Esto se muestra de la siguiente manera:

$$X_{ip} = X_{i(p-1)} + d.$$

$$X_{i^*p} = X_{i^*(p-1)} + d.$$

Paso 6: Para las rutas  $i$  e  $i^*$ , computamos los tiempos de llegas de todas las salidas ( $p=1, 2, \dots, \min(f_i, f_{i^*})$ ) para cada “posible” y “nuevo” nodo en la ruta:

$$T_{ip}^k = X_{ip} + t_{ik} \quad \forall k = 1, \dots, N \text{ presente en } i.$$

$$T_{i^*p}^k = X_{i^*p} + t_{i^*k} \quad \forall k = 1, \dots, N \text{ presente en } i^*.$$

Paso 7: Etiquetamos todos los demás nodos de esas rutas como “posibles”, imponiendo como “no posible” el nodo seleccionado ( $k^*$ )

Para comenzar el proceso del “procedimiento 1”, tenemos que tener en cuenta, que partimos de una variable denominada, *procedimiento* inicializada a cero. Con esta variable, podemos entrar en el grueso del cuerpo de este primer bloque, de tal forma que con ello, podemos darle valores a las siguientes variables:

- La distancia mínima ( $d_{\min}$ ).
- La distancia máxima ( $d_{\max}$ )
- La mínima frecuencia ( $\min\_freq[i]$ )
- El tiempo de llegada ( $arr\_time[i][j]$ )
- El tiempo de salida ( $dep\_time[i][j]$ )
- La parada de frecuencia ( $stop\_freq[i]$ )
- El chequeo de las rutas ( $check\_route[i]$ ).

En el caso de que en el comienzo del cuerpo del procedimiento 1, nos encontráramos con que la variable *procedimiento* es distinta de cero, toma el valor uno, tendríamos que pasar de forma directa a ejecutar el “procedimiento 2”.

#### Procedimiento 2.

Este procedimiento juega con los tiempos de salida, cuando el nodo seleccionado es posible. Para un nodo seleccionado “posible” tendremos algunas rutas, las cuales sus tiempos de comienzo estén ya listo para usarse en el procedimiento 1. Por eso, los tiempos de salida de las rutas que no están listas, se les asignarán la llegada simultanea con el conjunto de llegadas al nodo. Si no fueran posibles más asignaciones, el nodo se marcaría como nodo no posible.

Los siguientes “pasos” estarán transformados por este procedimiento:

Paso 1: En el nodo seleccionado  $k^*$ , marcado como “posible”, vemos cómo hemos fijado la ruta  $i^*$  como la que está atravesando dicho nodo, el cual contará con que todos los tiempos de salida ya han sido fijados en el Procedimiento 1. Por tanto, se fijan los tiempos de salida de la ruta  $i^*$  desde su origen hasta el nodo  $k^*$  como:

$$T_{i^*}^{k^*} = X_{i^*} + t_{ik^*} \quad \forall p = 1 \dots f_{i^*}$$

Paso 2: Para el resto de las rutas  $i$ , que están atravesando el nodo seleccionado, fijamos un valor  $p$  como el mínimo no asignado de frecuencia, donde  $p \in (1 \dots f_i)$ .

Para cada  $T_{i^*}^{k^*}$  se impone,  $T_{i^*}^{k^*}, T_{i^*}^{k^*}, \dots, T_{i^*}^{k^*}$ , fijado  $w = WT \min_{k^*}$

Paso 3: Para las rutas  $i$ , si  $0 \leq (T_{i^*}^{k^*} - w - t_{ik^*}) \leq Hmax_i + X_{i(p-1)}$

Para  $p=1$ , fijamos  $X_{i^*} = (T_{i^*}^{k^*} - w - t_{ik^*})$  y nos movemos hacia el paso 4.

Para  $p > 1$ , si  $(T_{i^*}^{k^*} - w - t_{ik^*}) - X_{i(p-1)} \geq Hmin_i$ , manteniendo  $X_{i^*} = (T_{i^*}^{k^*} - w - t_{ik^*})$ , y nuevamente nos movemos hacia el paso 4.

Si no, establecemos  $0 \leq (T_{i^*}^{k^*} - w - t_{ik^*}) \leq Hmax_i + X_{i(p-1)}$ , y nos trasladamos al paso 4.

Por lo demás, se fija  $w=w+1$ , y si  $w \leq WTmax_{k^*}$ , y repetimos el paso 3. Si no se fija  $T_{i^*}^{k^*} = T_{i^*}^{k^*}$ , manteniendo  $w=WTmin_{k^*}$ , y nuevamente repetimos el paso 3. Si no salimos.

Paso 4: Para las rutas  $i$ , si  $p < f_i$ , saltamos al paso 2. Etiquetando al nodo  $k^*$ , como “no posible”.

Paso 5: Para las rutas  $i$  que están pasando a través de  $k^*$ , cuyo tiempos de salida se han fijado en el Paso 3, se deberán de calcular los tiempos de llegadas para cada “posible” y “nuevo” nodo presente en la ruta. Este cálculo se realizará de la siguiente forma:

$$T_{i,p}^{k^*} = X_{i,p} + t_{ik^*} \quad \forall k = 1, \dots, N \text{ en la ruta } i.$$

Paso 6: Etiquetamos todos los restantes nodos “nuevos” de esas rutas, como “posibles”.

Procedimiento 3.

Este procedimiento chequea si hay alguna salida no asignada, que no haya sido creada por los dos primeros procedimientos.

1. Si hay sólo una salida no asignada de la ruta  $i$ , y listo  $p$  como el valor mínimo no asignado de frecuencia donde  $p \in (1 \dots f_i)$ .

Tenemos fijado los tiempos de salida, para el mínimo avance desde la última salida que se produce, para ello imponemos  $X_{i,p} = X_{i,(p-1)} + Hmin_i$ ,

2. Si hay más de una salida no asignada en diferentes rutas,

Identificar la ruta  $i$ , que está pasando a través del máximo número de nodos, rompe el bucle arbitrariamente.

Asignar las próximas salidas, usando el mínimo avance, desde la última salida, por ejemplo:  $X_{i,p} = X_{i(p+1)} + Hmin_i$ .

Todos los nodos, por los cuales se identifican las rutas, pasan a ser etiquetados como “posibles” otra vez.

Esto permitirá que el algoritmo pueda fijar adicionalmente, llegadas simultáneas como “posibles” nodos.

