

# Diseño de Horarios para Sincronizar Líneas de Buses en Múltiples Periodos de Planificación

O.J. Ibarra-Rojas  
Departamento de Ingeniería de Transporte y Logística  
Pontificia Universidad Católica de Chile  
oibarrar@uc.cl

Y.A. Rios-Solis  
Posgrado en Ingeniería de Sistemas  
Universidad Autónoma de Nuevo León  
yasmin.riosls@uanl.edu.mx

## RESUMEN

Presentamos el problema de Diseño de Horarios con Sincronización en Múltiples Periodos para determinar el tiempo de salida de un conjunto de viajes, maximizando el número de sincronizaciones que permiten una transferencia de pasajeros con tiempos de espera adecuados y evitan congestión de autobuses en algunos nodos de la red de transporte. Debe cumplirse regularidad de servicios en cada periodo de planificación y transiciones suaves entre diferentes periodos. Diseñamos una formulación y una Búsqueda de Vecindario Variable basada en operadores que usan propagación de restricciones. Obtenemos soluciones de calidad para instancias grandes y mostramos el beneficio de implementar el enfoque multiperiodo.

*Palabras claves:* diseño de horarios, sincronización, congestión vehicular, transferencia de pasajeros, metaheurística.

## ABSTRACT

We address the Multiperiod Synchronization Bus Timetabling problem that determines the departure times of a set of trips maximizing the number of synchronization events that allow well timed passenger transfers and avoid bus congestion at some nodes of the transit network. Constraints of the problem are regular services in each planning period and smooth transitions between different periods. We propose a mathematical formulation and a Variable Neighborhood Search algorithm that is based on operators using constraint propagation. We empirically obtain high quality feasible solutions for real size instances and show the benefit of implement the multiperiod approach.

*Keywords:* timetabling, synchronization, bus congestion, passenger transfers, variable neighborhood search.

## 1. INTRODUCCIÓN

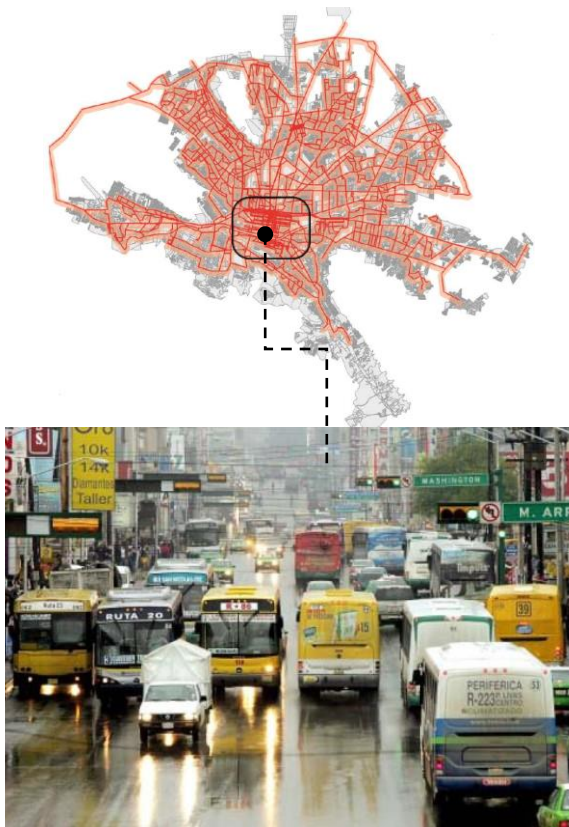
La eficiencia del transporte público depende de diversos elementos como; infraestructura, políticas gubernamentales, tecnología disponible, planificación y operación diaria del mismo, entre otros. En particular, la planificación es un proceso bastante complejo que se divide en subproblemas como: diseño de líneas de transporte, cálculo de frecuencias de paso, diseño de horarios, asignación de vehículos y asignación de personal (Desaulniers y Hickman, 2007). El enfoque más común es resolver estos problemas de forma secuencial por lo que es de gran importancia contar con métodos de solución eficientes para cada uno de los subproblemas a fin de obtener una mayor calidad en la solución global del proceso de planificación.

En este estudio, nos enfocamos en el diseño de horarios mediante técnicas de investigación de operaciones para mejorar el sistema de transporte de Monterrey, México, el cual es manejado por varias compañías haciendo que la competencia por obtener una mayor demanda está siempre presente. Este hecho lleva a tener una red de transporte con más de 400 líneas, donde una gran cantidad líneas tienen tramos de rutas en común causando una gran congestión vehicular.

El panel izquierdo de la Figura 1 ilustra la red de transporte de Monterrey, el recuadro resaltado en el mapa muestra el centro de la ciudad donde convergen una gran cantidad de líneas. El panel derecho de la Figura 1 muestra la avenida Juárez localizada en el centro de Monterrey, donde se puede observar la congestión de autobuses de diferentes líneas.

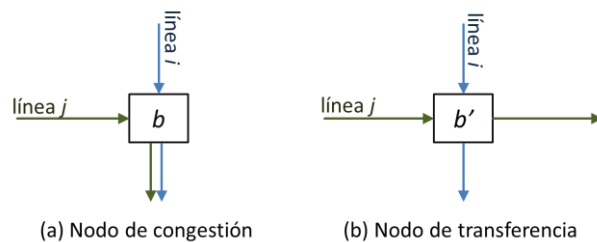
Considerando estas características, optimizar la interacción de líneas se vuelve una meta bastante razonable a fin de reducir la saturación de autobuses en paraderos (y congestión en tramos de rutas comunes) así como mejorar la calidad e imagen del sistema de transporte. Las características del sistema son las siguientes: se proveen servicios regulares ya que el horario es solo para las agencias que proveen el servicio; la transferencia de pasajeros con tiempos de espera razonables es necesaria en zonas como el centro de la ciudad donde una gran cantidad de pasajeros debe transferir a otras líneas para llegar a su destino final; se desea disminuir la congestión vehicular causada por la saturación de autobuses de diferentes líneas en paraderos y a lo largo de tramos de rutas comunes. Proponemos controlar la transferencia de pasajeros y la separación de autobuses en paraderos mediante la separación entre las llegadas de dos viajes a un nodo común.

La Figura 2 ilustra ambos casos, el panel izquierdo muestra el caso de dos líneas que comparten un tramo de ruta empezando en un punto  $b$ , mientras que el panel derecho muestra un punto  $b$  donde es necesaria la transferencia de pasajeros.



**FIGURA 1:** Red de Transporte de Monterrey y Saturación de Líneas en el Centro de la Ciudad

Fuente: Ibarra-Rojas (2013)



**FIGURA 2:** Nodo de Congestión de Autobuses y Nodo de Transferencia de Pasajeros para Líneas  $i$  y  $j$ .

Para definir un evento de sincronización que reduce la congestión de autobuses, buscamos una separación lo suficientemente grande entre la llegada de dos autobuses a un nodo de congestión  $b$  evitando que los viajes vayan muy cercanos durante del tramo en común. Por otro lado, se define una sincronización que favorece la transferencia de pasajeros cuando existe un tiempo de separación entre llegadas de dos autobuses a un nodo de transferencia  $b'$ , tal que se permita a los pasajeros transferir de la línea  $i$  a la línea  $j$  en tiempos de espera razonables. Un enfoque común para reducir la variabilidad en parámetros, es dividir un día en periodos de planificación más pequeños donde el sistema de transporte tenga un “comportamiento” menos variable a fin de usar formulaciones deterministas y unir las soluciones correspondientes. Sin embargo, aún en un periodo de planificación pequeño, por ejemplo de 2 horas, puede existir una gran variabilidad en tiempos de recorrido y los periodos de planificación no tienen que ser necesariamente los mismos para todas las líneas.

Basados en lo anterior, nuestro problema consiste en *determinar los horarios en múltiples periodos de planificación para cada línea garantizando regularidad de servicios y maximizando el número de sincronizaciones a fin de reducir la congestión de autobuses y favorecer la transferencia de pasajeros a lo largo del día en ciertos puntos de la red de transporte.*

## 2. LITERATURA RELACIONADA

El objetivo más común basado en transferencia de pasajeros, es la minimización de los tiempos de espera para transferir de una línea a otra (Cevallos y Zhao, 2006; Chakroborty *et al.*, 1995; Jansen y Nielsen, 2002; Wong *et al.*, 2008). En estos estudios, es común considerar un periodo específico de planificación y una política de servicios regulares, es decir, los tiempos de salida de los viajes están separados por un tiempo constante llamado cabecera. Dada la intratabilidad del problema, la mayoría de estos estudios implementan algoritmos heurísticos a fin de obtener soluciones rápidamente. Estudios como los de Bookbinder y Désilets (1992) y Kwan y Chang (2008) se basan también en la minimización de tiempos de espera en transferencias pero, considerando el caso de trenes. Aunque estos estudios son aplicables a algunas redes de transporte público, existen diferencias entre la operación de trenes y de autobuses que impiden aplicar otros estudios a nuestro caso. Entre estas diferencias están las restricciones duras de separación de trenes en estaciones de intersección de líneas para evitar colisiones o los tipos de horario de servicio como periódico y “clockface service” (tiempos de salida que se repiten cada cierto periodo de tiempo, como por ejemplo, una hora).

Aunque, la minimización de tiempos de espera es un objetivo razonable, Schröder y Solchenbach (2006) resaltan la necesidad de implementar otras medidas ya que la minimización de tiempos de espera puede llevar a transferencias riesgosas si tomamos en cuenta que existe variabilidad en los tiempos de recorrido. Para mitigar este hecho, los autores proponen una función objetivo para redes de tránsito (aplicable a buses o trenes) basada en ventanas de tiempo de espera para definir casos como casi transferencia, transferencia no aceptable y transferencia con tiempos de espera pequeños, aceptables y largos. Otro enfoque es definir una sincronización en términos de la separación entre tiempos de llegada de dos autobuses a un paradero, lo cual abre la puerta a formulaciones con una flexibilidad que permite modelar diversos casos. Por ejemplo, Ceder y Tal (2001) y Ceder *et al* (2001) definen el problema de diseño de horarios a fin de maximizar el número de arribos simultáneos a un nodo en común. El problema cuenta con flexibilidad y regularidad en los servicios al considerar tiempos mínimos y máximos de cabecera para cada línea en lugar de un solo tiempo de cabecera. Debido a la intratabilidad del problema, se proponen algoritmos heurísticos para resolver el problema. Este enfoque es extendido en Eranki (2004) redefiniendo la sincronización como el evento de dos autobuses llegando a un punto en común con tiempo de separación entre una ventana de tiempo.

Recientemente, Ibarra-Rojas y Rios-Solis (2012) adecuan los modelos de Ceder (2001) y Eranki (2004) para maximizar el número de sincronizaciones de dos autobuses que permite transferencia de pasajeros o separa dos vehículos en puntos de congestión garantizando regularidad de servicios a lo largo de un periodo de planificación. Los autores demuestran que su problema y los de Ceder (2001) y Eranki (2004) son NP-duros. Después, en Ibarra-Rojas *et al* (2012) se definen formulaciones matemáticas que permiten resolver en minutos instancias de hasta 200 líneas de transporte y 40 nodos de sincronización utilizando un solver comercial.

Aunque el problema presentado en Ibarra-Rojas *et al* (2012) aborda nuestro caso de estudio, dicho problema está basado en un solo periodo de planificación.

Como veremos a lo largo de este trabajo, considerar diferentes periodos de planificación para cada línea y sincronización entre viajes pertenecientes a diferentes periodos de planificación, permite modelar el problema de una mejor forma y la calidad de las soluciones supera ampliamente el enfoque de unir horarios de periodos independientes.

### 3. DISEÑO DE HORARIOS CON SINCRONIZACIÓN MULTIPERIODO

Nuestro problema de Diseño de Horarios con Sincronización en Múltiples Periodos de Planificación (DHSM) consiste en determinar los tiempos de salida para los viajes de todas las líneas en los diferentes periodos de planeación tal que la separación entre viajes consecutivos este dentro de un mínimo y máximo tiempo de cabecera para cada periodo y se maximice el número de sincronizaciones que permiten una transferencia de pasajeros o evitan la congestión de buses en nodos en común.

DHSM es una generalización del problema de un solo periodo presentado en Ibarra-Rojas (2012), y tal como sucede en este último estudio, una sincronización está definida como el evento donde dos autobuses llegan a un punto en común con un tiempo de separación entre una ventana de tiempo llamada ventana de sincronización.

La gran diferencia, es el hecho de considerar que cada línea tiene periodos de planeación específicos donde es posible estimar de forma más precisa los tiempos de recorrido.

Por ejemplo, una línea  $i$  puede tener poca variabilidad en los parámetros durante el periodo de 7:00 am a 7:30 am, mientras que para otra línea  $j$ , la variabilidad es menor durante el periodo de 7:20 am a 8:00 am. Si es posible definir la sincronización de viajes pertenecientes a diferentes periodos de planeación, se tiene un problema muy flexible que puede considerar una gran cantidad de periodos de planificación lo suficientemente pequeños, tal que sea representativo considerar parámetros deterministas como los tiempos de recorrido, que también se ven favorecidos por la implementación de corredores exclusivos en algunos sistemas de transporte. Similarmente, podrían considerarse sistemas intermodales respetando el supuesto de conocimiento de tiempos de recorrido por ejemplo, el metro. A continuación, presentamos la formulación matemática del problema y la forma en la que definimos las ventanas de sincronización que permiten implementar el enfoque multiperiodo.

#### 3.1 Formulación Lineal Entera-Mixta para DHSM

La red de transporte está representada por un conjunto  $I$  de rutas y un conjunto  $B$  de nodos de sincronización. Para toda línea  $i \in I$ , definimos un conjunto  $J(i) \subseteq I$  de líneas con las cuales  $i$  puede sincronizarse. El conjunto  $B^{ij}$  representa los nodos de sincronización para una pareja de líneas  $(i, j \in J(i))$  y  $S^i$  denota los periodos de planificación para cada línea  $i$ . Los parámetros del problema son los siguientes:  $d_s^i$ , que representa el final del periodo  $s$  y el inicio del periodo  $s + 1$  para la línea  $i$ ; el número de viajes  $f_s^i$  necesarios para satisfacer la demanda de la línea  $i$  en el periodo  $s$ ;  $f^i$  el número de viajes de línea  $i$  en todo el día de planeación; cantidad de minutos  $t_p^{ib}$  que tarda el  $p$ -ésimo viaje de la línea  $i$  (denotado por  $i(p)$ ) en llegar al nodo  $b$  desde el origen; finalmente,  $[w_{pq}^{ijb}, W_{pq}^{ijb}]$  representa la ventana de sincronización

para los viajes  $i(p)$ ,  $j(q)$  con  $j \in J(i)$  y nodo  $b \in B^{ij}$ . Los tiempos de salida deben garantizar servicios regulares en la forma de viajes consecutivos casi igualmente separados. Entonces, para toda línea  $i$  y periodo  $s \in S^i$ , consideramos tiempos mínimos y máximos de cabecera  $h_s^i = \mu_s^i - \Delta_s^i$  y  $H_s^i = \mu_s^i + \Delta_s^i$ , respectivamente, donde  $\mu_s^i = \frac{d_s^i - d_{s-1}^i}{f_s^i}$  es el tiempo de cabecera que garantiza viajes igualmente espaciados y el parámetro  $\Delta_s^i$  representa la flexibilidad del horario. Las principales variables de decisión son los tiempos de salida desde el origen para cada viaje  $i(p)$  denotado por variables reales  $X_p^i$ . Entonces, definimos las variables binarias  $Y_{pq}^{ijb}$  que toman el valor de 1 si  $i(p)$  llega primero a nodo  $b$  y sincroniza con  $j(q)$  y son 0 en otro caso. Utilizando estas variables de decisión, la formulación de DHSM está dada como sigue.

$$\max F_{DHSM}(Y) = \sum_{i \in I} \sum_{j \in J(i)} \sum_{b \in B^{ij}} \sum_{p=1}^{f_i^i} \sum_{q=1}^{f_j^j} Y_{pq}^{ijb} \quad (1)$$

Sujeto a:

$$h_s^i \leq X_{p+1}^i - X_p^i \leq H_s^i \quad \forall i \in I, p = first(s), \dots, last(s) - 1, s \in S \quad (2)$$

$$d_{s-1}^i + \frac{h_s^i}{2} \leq X_{first(s)}^i \leq d_{s-1}^i + \frac{H_s^i}{2} \quad \forall i \in I, s \in S \quad (3)$$

$$d_s^i - \frac{H_s^i}{2} \leq X_{last(s)}^i \leq d_s^i - \frac{h_s^i}{2} \quad \forall i \in I, s \in S \quad (4)$$

$$X_q^j + t_q^{jb} - (X_p^i + t_p^{ib}) \geq w_{pq}^{ijb} - M(1 - Y_{pq}^{ijb}) \quad \forall i(p), j(q), j \in J(i), b \in B^{ij} \quad (5)$$

$$X_q^j + t_q^{jb} - (X_p^i + t_p^{ib}) \leq w_{pq}^{ijb} + M(1 - Y_{pq}^{ijb}) \quad \forall i(p), j(q), j \in J(i), b \in B^{ij} \quad (6)$$

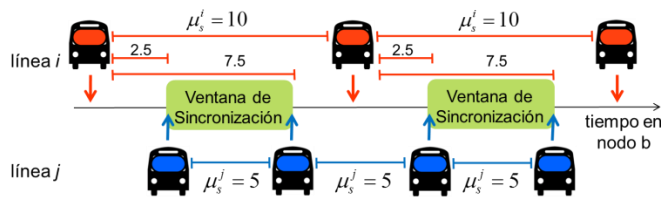
La función objetivo  $F_{DHSM}(Y)$  maximiza el número de sincronizaciones que se calculan a partir de los tiempos de salida  $X$ . Las restricciones (2) garantizan la regularidad de servicios para cada línea  $i$  en cada periodo de planeación  $s \in S^i$ . Para modelar una transición suave entre periodos de planeación, las ecuaciones (3) y (4) obligan a que la separación entre la salida del último viaje de un periodo  $s$  (denotado como  $last(s)$ ) y el primer viaje del siguiente periodo (denotado  $first(s+1)$ ) este cerca del promedio de las cabeceras del periodo  $s$  y  $s+1$ . Por último, las restricciones (5) y (6) permiten activar una variable  $Y_{pq}^{ijb}$  si la diferencia entre el tiempo de arribo de los viajes  $j(q)$  e  $i(p)$  al nodo  $b$  esta dentro de la ventana de sincronización  $[w_{pq}^{ijb}, W_{pq}^{ijb}]$ . El valor del parámetro  $M$  debe ser lo suficientemente grande para acotar la diferencia entre las llegadas de  $i(p)$  y  $j(q)$  al nodo  $b$ .

En el caso de transferencia de pasajeros, la ventana de sincronización puede definirse en base tiempos de espera razonables desde el punto de vista del pasajero. En el caso de evitar congestión de autobuses, las ventanas de sincronización dependen de los viajes  $i(p)$  y  $j(q)$ . Por ejemplo, consideremos dos líneas  $i$  y  $j$  con  $\mu_s^i = 10$  y  $\mu_s^j = 5$  en un periodo de planificación  $s$ . En este caso, podemos definir la ventana de sincronización  $[w_{pq}^{ijb}, W_{pq}^{ijb}]$  como

$$[0.5 \min(\mu_s^i, \mu_s^j), \max(\mu_s^i, \mu_s^j) - 0.5 \min(\mu_s^i, \mu_s^j)] = [2.5, 7.5],$$

lo cual permite que dos viajes de la línea  $j$  arriben a  $b$  entre el arribo de dos viajes consecutivos de la línea  $i$  (ver Figura 3).

En otras palabras, al maximizar el número de sincronizaciones entre estas dos líneas, se logra una armonización entre los arribos de estas líneas en los nodos de congestión.



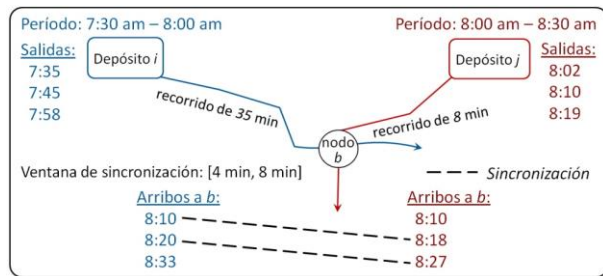
**FIGURA 3: Diseño de Ventanas de Sincronización para Armonizar los Arribos de Viajes de dos Líneas a un Nodo de Sincronización**  
Fuente: Ibarra-Rojas (2013)

En general, para modelar la sincronización entre dos viajes  $i(p)$  y  $j(q)$  pertenecientes a diferentes periodos de planificación, la ventana de sincronización puede expresarse en base a las cabeceras de los viajes en cuestión, es decir,

$$[w_{pq}^{ijb}, W_{pq}^{ijb}] = [0.5 \min(\hat{\mu}_p^i, \hat{\mu}_q^j), \max(\hat{\mu}_p^i, \hat{\mu}_q^j) - 0.5 \min(\hat{\mu}_p^i, \hat{\mu}_q^j)]$$

donde  $\hat{\mu}_p^i$  y  $\hat{\mu}_q^j$  se definen como sigue. Si  $i(p) = last(s)$  para algún periodo  $s$ , la separación entre el viaje  $i(p)$  y el siguiente viaje de la línea  $i$  es un valor cercano al promedio de las cabeceras del periodo  $s$  y  $s + 1$ . Entonces, definimos  $\hat{\mu}_p^i = \frac{\mu_s^i + \mu_{s+1}^i}{2}$ . En cualquier otro caso, usamos  $\hat{\mu}_p^i = \mu_s^i$ , donde  $s$  es el periodo al que pertenece  $i(p)$ . En el caso de  $j(q)$ , si  $j(q) = first(s')$  o  $j(q) = last(s' - 1)$ , el viaje  $i(p)$  pudiera sincronizar con viajes pertenecientes a  $s' - 1$ ,  $s'$  ó ambos. Para permitir lo anterior, definimos  $\hat{\mu}_q^j = \min\left(\mu_{s'-1}^j, \frac{\mu_{s'-1}^j + \mu_{s'}^j}{2}, \mu_{s'}^j\right)$ . En otro caso, usamos  $\hat{\mu}_q^j = \mu_{s'}^j$ , donde  $s'$  es el periodo al que pertenece  $j(q)$ .

Usando DHSM podemos modelar el caso donde cada línea puede tener intervalos de planificación lo suficientemente pequeños tal que sea representativo asumir que los tiempos de recorrido son deterministas. Por ejemplo, la Figura 4 muestra dos líneas cuyos viajes pertenecen a diferentes periodos de planificación (de 30 minutos). El primer y segundo viaje de la línea  $i$  están sincronizados con el segundo y tercer viaje de la línea  $j$ , respectivamente, ya que el tiempo de separación entre arribos está dentro de la ventana [4, 8]. Estas sincronizaciones serían ignoradas implementando el enfoque de resolver horarios para periodos independientes y unir las soluciones para obtener un horario de todo el día.



**FIGURA 4: Sincronización entre Viajes Pertenecientes a Diferentes Periodos de Planificación**  
Fuente: Ibarra-Rojas (2013)

El número de variables en DHSM crece considerablemente comparado con el enfoque de un solo periodo de planeación y además, DHSM es NP-duro ya que contiene como un caso particular, al problema NP-duro de diseño de horarios presentado en Ibarra-Rojas y Rios-Solis (2012).

En particular, experimentación preliminar implementando el solver de CPLEX 12 limitado a 8 horas de tiempo de ejecución para resolver instancias pequeñas lleva a una lenta convergencia y soluciones de calidad inaceptable (ver Ibarra-Rojas 2013).

Entonces, el diseño de algoritmos de solución eficientes para DHSM es de vital importancia para implementar enfoques secuenciales o integrales a fin de resolver el problema global de planificación. En este estudio, proponemos un algoritmo de Búsqueda de Vecindario Variable (VNS por su nombre en inglés Variable Neighborhood Search) para obtener soluciones de buena calidad para el DHSM de forma eficiente, lo cual no es posible con solvers comerciales. La idea básica de un algoritmo VNS es el explorar estratégicamente diferentes vecindades de una solución factible para obtener una solución que sea un óptimo local respecto a todos los vecindarios, con el objetivo de que la solución sea de mayor calidad que la que se pueda encontrar explorando los vecindarios independientemente (Hansen and Mladenovic, 2003). Para generar los diferentes tipos de vecindarios utilizamos el concepto de propagación de restricciones descrito en la siguiente sección.

### 3.2 Propagación de Restricciones

La propagación de restricciones puede definirse como cualquier razonamiento que prohíba explícitamente valores o combinaciones de valores para algunas variables de un problema de optimización a fin de garantizar factibilidad (Bessiere, 2006). En este estudio, la propagación de restricciones está basada en la regularidad de servicio y nos permitirá definir la región factible de los tiempos de salida  $X_p^i$  e identificar cuáles son los valores de los tiempos de salida de dos viajes que garantizan su sincronización. Por ejemplo, supongamos que tenemos el valor de una variable  $X_{p'}^i$  para un viaje  $i(p')$  en un periodo  $s$ . Sabemos que el tiempo de separación entre viajes consecutivos de la línea  $i$  en el periodo  $s$  debe estar dentro del intervalo  $[h_s^i, H_s^i]$ .

Entonces, podemos definir un intervalo de factibilidad  $D_p^i$  para cualquier otro viaje  $i(p)$  en el periodo  $s$  con  $p > p'$  como  $[X_{p'}^i + (p - p')h_s^i, X_{p'}^i + (p - p')H_s^i]$ .

Análogamente, para un viaje  $i(p)$  en el periodo  $s$  con  $p' > p$ ,  $D_{p'}^i$  esta dado por  $[X_p^i - (p' - p)H_s^i, X_p^i - (p' - p)h_s^i]$ .

Mientras más valores de variables se conozcan, más preciso es el intervalo de factibilidad  $D_p^i$ . Si no tenemos valores de los tiempos de salida, podemos utilizar las restricciones que acotan los tiempos de salida del primer y último viaje de cada línea en cada periodo de planeación. Por un lado,  $d_{s-1}^i + \frac{h_s^i}{2}$  y  $d_{s-1}^i + \frac{H_s^i}{2}$  son la cota inferior y superior de la variable  $X_{first(s)}^i$ , respectivamente. Por otro lado,  $d_s^i - \frac{h_s^i}{2}$  y  $d_s^i - \frac{H_s^i}{2}$  son la cota inferior y superior de la variable  $X_{last(s)}^i$ , respectivamente. Entonces,  $D_p^i$  puede calcularse como  $[d_{s-1}^i + \frac{h_s^i}{2} + (p - first(s))h_s^i, d_{s-1}^i + \frac{H_s^i}{2} + (p - first(s))H_s^i] \cap [d_s^i - \frac{h_s^i}{2} - (last(s) - p)H_s^i, d_s^i - \frac{H_s^i}{2} - (last(s) - p)h_s^i]$ .

Desplazando el intervalo factible de los tiempos de salida  $X_p^i$  de cada viaje  $i(p)$  por  $t_{pb}^i$  unidades de tiempo, podemos determinar un intervalo de llegada factible  $A_p^{ib}$  para cada uno de los nodos de sincronización  $b$ . Entonces, la ventana factible de arribo de  $i(p)$  a  $b$  está dada como  $A_p^{ib} = [\min(D_p^i) + t_{pb}^i, \max(D_p^i) + t_{pb}^i]$  donde las funciones min y max, representan el valor mínimo y máximo del intervalo en el argumento, respectivamente. Más aún, utilizando la información de las ventanas de sincronización, podemos definir un intervalo

$S_{pq}^{ijb} = [\min(A_p^{ib}) + w_{pq}^{ijb}, \max(A_p^{ib}) + W_{pq}^{ijb}]$  donde, si un viaje  $j(b)$  llega a  $b$  dentro de dicho intervalo, es posible que se sincronice con  $i(p)$ . Un resultado directo de la definición de los intervalos de salida, arribo y sincronización, está dado por el siguiente teorema cuya demostración es trivial siguiendo la propagación de restricciones para el caso de un solo periodo mostrada en Ibarra-Rojas y Rios-Solis (2012).

**Teorema 1:** Para todos los viajes  $i(p)$ ,  $j(q)$  con  $j \in J(i)$  y nodo de sincronización  $b \in B^{ij}$ ,  $Y_{pq}^{ijb} = 1$  es infactible, si y sólo si,  $S_{pq}^{ijb} \cap A_q^{jib} = \emptyset$ .

El teorema anterior identifica si es posible sincronizar dos viajes. Sin embargo, también es posible encontrar intervalos factibles para los tiempo de salida dos viajes  $i(p)$  y  $j(q)$  tal que se garantice su sincronización en un nodo  $b$ . Usamos el símbolo “ $\bar{\cdot}$ ” en los intervalos, ya sean de salida, arribo o sincronización, que se obtienen con una solución parcial, es decir, con información de los tiempos de salida. Entonces, la sincronización de dos viajes  $i(p)$  y  $j(q)$  en un nodo  $b$  se garantiza cuando se cumple lo siguiente.

$$X_q^j \in \bar{D}_{qp}^{jib} = [\min(\bar{S}_{pq}^{ijb} \cap \bar{A}_q^{jib}) - t_q^{jb}, \max(\bar{S}_{pq}^{ijb} \cap \bar{A}_q^{jib}) - t_q^{jb}] \cap \bar{D}_q^j$$

$$X_p^i \in \bar{D}_{pq}^{ijb} = [X_q^j + t_q^{jb} - t_p^{ib} - W_{pq}^{ijb}, X_q^j + t_q^{jb} - t_p^{ib} - w_{pq}^{ijb}] \cap \bar{D}_p^i$$

Por la definición de los diferentes intervalos de salida,  $\bar{D}_{pq}^{ijb} \subseteq \bar{D}_p^i \subseteq D_p^i$ . En resumen, los intervalos definidos anteriormente y el Teorema 1, son herramientas que pueden ser utilizadas para definir y explorar la región factible del problema.

#### 4. ALGORITMO DE BÚSQUEDA DE VECINDARIO VARIABLE

Dada la intratabilidad del problema, proponemos implementar una Búsqueda de Vecindario Variable (VNS). Dicho algoritmo, está basado en la exploración estratégica de diferentes estructuras de vecindarios  $N_1, \dots, N_{k_{max}}$  de una solución factible  $X$ . El Algoritmo 1 muestra nuestro VNS, el cual toma una solución inicial y fija el primer vecindario  $k$  a explorar (paso 2). Después, se hace una exploración de los diferentes vecindarios (paso 3) donde en cada iteración, se toma un vecino aleatorio del vecindario  $k$ , para después implementar una búsqueda local. Si la nueva solución tiene un mejor valor de la función objetivo  $F_{DHSM}(Y)$ , se toma como la solución actual y se sigue explorando el vecindario  $k$ , en caso contrario, se explora el siguiente vecindario (pasos 4-9). El algoritmo se detiene hasta alcanzar un número  $iter\_limit$  de iteraciones sin mejorar la función objetivo (paso 1).

##### Algoritmo 1: VNS

**Entrada:** Solución factible  $X$

**Salida:** Solución factible  $X^*$

```

1: while iteraciones sin mejorar  $F_{DHSM}(Y) < iter\_limit$  do
2:    $k = 1$ 
3:   while  $k \leq k_{max}$  do
4:     toma un vecino aleatorio  $X' \in N_k(X)$ ,  $X'^* \leftarrow busquedalocal(X')$ 
5:     if  $F_{DHSM}(Y'^*) < F_{DHSM}(Y)$  then
6:        $k \leftarrow k + 1$ 
7:     else
8:        $X \leftarrow X'^*$ 
9:     end if
10:  end while
11: end while

```

Como podemos notar en el Algoritmo 1. Es necesario contar con una solución inicial factible y diferentes vecindarios. En las siguientes subsecciones abordamos cada uno de estos aspectos.

#### 4.1 Solución Inicial

Diseñamos un algoritmo llamado *inicial()* que construye un ranking de las variables de sincronización como sigue. Asumamos que  $Y_{pq}^{ijb} = 1$  para ciertos viajes  $i(p)$ ,  $j(q)$  y nodo  $b$ . Entonces, los tiempos de salida  $X_p^i$  y  $X_q^j$  de  $i(p)$  y  $j(q)$  deben estar dentro de intervalos específicos  $\bar{D}_p^i$  y  $\bar{D}_q^j$ , respectivamente. Esto resulta en una nueva región factible para los tiempos de salida y por Teorema 1, existe un conjunto  $E_{pq}^{ijb}$  de variables de sincronización que son imposibles de activar asumiendo  $Y_{pq}^{ijb} = 1$ . En otras palabras,  $E_{pq}^{ijb}$  es el conjunto de sincronizaciones “bloqueadas” al asumir  $Y_{pq}^{ijb} = 1$ . En base a lo anterior, podemos hacer un listado LISTSYNC de las variables de sincronización  $Y_{pq}^{ijb}$  ordenado crecientemente respecto al número de sincronizaciones bloqueadas ( $|E_{pq}^{ijb}|$ ). Después, podemos construir la solución inicial activando las sincronizaciones en esta lista ordenada como ilustra el Algoritmo 2.

##### Algoritmo 2: inicial()

**Entrada:** instancia de DHSM

**Salida:** Solución factible  $X$

```

1: while LISTSYNC no vacía do
2:   toma primer elemento  $Y_{pq}^{ijb}$  de LISTSYNC
3:    $X_q^j \leftarrow rand(\bar{D}_{qp}^{jib})$ ,  $X_p^i \leftarrow rand(\bar{D}_{pq}^{ijb})$ 
4:   LISTSYNC = LISTSYNC -  $\{Y_{pq}^{ijb}\} \cup E_{pq}^{ijb}$ 
5: end while
6: for cada viaje  $i(p)$  sin tiempo de salida asignado do
7:    $X_p^i \leftarrow rand(\bar{D}_p^i)$ 
8: end for

```

#### 4.2 Vecindarios para DHSM

Un vecindario  $N(X)$  de una solución  $X$  se define como todas las soluciones que se obtienen de implementar un operador bien definido dicha solución. En nuestro caso, estamos interesados en el operador definido por el desplazamiento de un viaje o una línea para forzar algunas sincronizaciones. Por ejemplo, consideremos una solución factible  $X$  y sea  $X_p^i$  el tiempo de salida del viaje  $i(p)$ .

Si conocemos el tiempo de salida  $X_q^j$ , podemos identificar si es posible la sincronización de  $i(p)$  con  $j(q)$  en un nodo  $b$  utilizando el Teorema 1. En caso afirmativo, definimos el operador  $shift(X, i(p), j(q), b)$  como un desplazamiento aleatorio del tiempo de salida  $X_p^i$  para forzarlo a estar dentro del intervalo  $\bar{D}_{pq}^{ijb}$  y garantizar  $Y_{pq}^{ijb} = 1$ , es decir,  $shift(X, i(p), j(q), b) = rand(\bar{D}_{pq}^{ijb})$

donde  $rand(\bar{D}_{pq}^{ijb})$  representa la generación de un número aleatorio en el intervalo  $\bar{D}_{pq}^{ijb}$ . Basándonos en este operador, definimos a  $N_{trip,i}(X)$ , como el conjunto de todas las soluciones obtenidas de implementar a una solución  $X$  el operador  $shift(X, i(p), j(q), b)$  para todos los viajes  $i(p)$ ,  $j(q)$  con  $j \in J(i)$  y nodo  $b \in B^{ij}$ .

Análogamente, definimos el operador  $shift(X, j(q), i(p), b)$  para desplazar el tiempo de salida del viaje  $j(q)$  dentro de  $\widehat{D}_{qp}^{ib}$  y garantizar  $Y_{pq}^{ib} = 1$ . También, definimos el operador  $shift(X, [i(p) \sim j(q)], b)$  el cual desplaza a ambos viajes  $i(p)$  y  $j(q)$  para garantizar su sincronización. De igual forma, definimos los vecindarios  $N_{trip\_j}(X)$  y  $N_{trip\_ij}(X)$  basados en los operadores  $shift(X, j(q), i(p), b)$  y  $shift(X, [i(p) \sim j(q)], b)$ , respectivamente.

Ahora, implementamos sucesivamente los desplazamientos de un solo viaje para definir el desplazamiento de toda una línea o una pareja de líneas para tratar de alcanzar más sincronizaciones. Por ejemplo, el operador  $shift(X, i, j, b)$  representa la sucesiva implementación de  $shift(X, i(p), j(q), b)$  para las parejas de viajes  $(i(1), j(1)), (i(1), j(2)), \dots, (i(f^i), j(f^j))$ . Análogamente, definimos los operadores  $shift(X, j, i, b)$  y  $shift(X, [i \sim j], b)$  resultando en los vecindarios  $N_{line\_i}(X)$ ,  $N_{line\_j}(X)$  y  $N_{line\_ij}(X)$ .

### 4.3 Búsquedas locales para DHSM

Una búsqueda local escoge una solución de un vecindario en base a una medida de desempeño y una regla de exploración. Por ejemplo, dada una solución  $X$  y un vecindario  $N(X)$ , puede implementarse un enfoque de primer mejora encontrada, la cual toma la primer solución  $X' \in N(X)$  que tenga un mejor valor de la función objetivo.

En nuestro caso, implementamos dicho procedimiento para explorar los diferentes tipos de vecindarios, dando como resultado las siguientes búsquedas locales:

$localsearch_{trip\_i}()$ ,  $localsearch_{trip\_j}()$ ,  $localsearch_{trip\_ij}()$ ,  
 $localsearch_{line\_i}()$ ,  $localsearch_{line\_j}()$  y  $localsearch_{line\_ij}()$ .

Por ejemplo, el Algoritmo 3 describe  $localsearch_{trip\_i}()$ , donde podemos ver que se hace una exploración lexicográfica de las parejas de viajes a fin de identificar si su sincronización mejora el valor de la función objetivo, y en caso afirmativo se toma la nueva solución como la actual.

**Algoritmo 3:**  $localsearch_{trip\_i}()$ .

**Entrada:** Solución factible  $X$

**Salida:** Solución factible  $X^*$

```

1: while  $F_{DHSM}(Y)$  sea mejorada do
2:   for cada pareja de líneas  $(i, j \in J(i))$  y nodo  $b \in B^j$  do
3:     for  $p = 1$  hasta  $p = f^i$  do
4:       for  $q = 1$  hasta  $q = f^j$  do
5:          $X' \leftarrow shift(X, i(p), j(q), b)$ 
6:          $X' \leftarrow X$ , si  $F_{DHSM}(Y') > F_{DHSM}(Y)$ 
7:       end for
8:     end for
9:   end for
10: end for
11: end while

```

A diferencia de los estudios de van den Heuvel *et al* (2008) y Guihaire y Hao (2010) en donde se implementan búsquedas locales basadas en desplazamiento de viajes por cantidades de tiempo conocidas a priori (por ejemplo, 1 min, 3 min, 5 min y 10 min), nuestros operadores de desplazamiento utilizan la propagación de restricciones para definir la región factible del problema y explorarla de forma eficiente identificando potenciales sincronizaciones y sus respectivos intervalos de sincronización antes de realizar los desplazamientos.

En la siguiente sección veremos que nuestra metodología de solución es capaz de encontrar soluciones de buena calidad y mostraremos la ventaja de utilizar el enfoque multiperiodo para la sincronización de líneas.

## 5. RESULTADOS EXPERIMENTALES

Programamos nuestros algoritmos usando C++ y todos los experimentos fueron realizados en un equipo iMac OS X con procesador Intel Core 2 Duo, 3.06 GHz, 4 GB RAM. Tanto la construcción inicial como el algoritmo VNS tienen decisiones que se toman al azar. Entonces, implementamos un enfoque multiarranque (denotado como MVNS) que consiste en hacer 10 ejecuciones que construyen una solución usando  $inicial()$  y la mejoran utilizando VNS con un criterio de parada de  $iter\_limit = 20$ . Después de las 10 ejecuciones, se toma la mejor solución obtenida para nuestro análisis.

### 5.1 Eficiencia del Algoritmo

Para medir la eficiencia del algoritmo propuesto, generalizamos el diseño de las instancias presentado en Ibarra-Rojas *et al* (2012), el cual está basado en observaciones de un experto en la gestión del sistema de transporte de México. El tamaño de las instancias está representado por el número de líneas  $|I|$  y el número de nodos de sincronización  $|B|$ . Todas las instancias consideran 6 periodos de planificación de 240 min. Los parámetros son generados aleatoriamente entre los siguientes intervalos: las cantidades de viajes  $f_s^i \in [13, 18]$ ; el parámetro de flexibilidad cumple  $\frac{\Delta_i}{\mu_s} 100\% \in [15\%, 25\%]$ ; los tiempos de recorrido desde origen a nodos de sincronización  $b$  cumplen que  $t_{pb}^i \in [20, 60]$  y  $t_{pb}^i = t_{p'b}^i$  para dos viajes  $p$  y  $p'$  en el mismo periodo de planificación; el número de parejas de líneas a sincronizar en cada nodo está entre  $[1, 7]$ . El nombre y tamaño de las instancias está representado en la Tabla 1.

**TABLA 1: Diferentes Tipos de Instancias Propuestas en Ibarra-Rojas *et al* (2013)**

Tipo de Instancia	A1	A2	A3	A4
( I ,  B )	(10, 1)	(50, 5)	(100, 10)	(200, 20)

Se generaron 30 instancias de tipo y dichas instancias tienen la particularidad que los parámetros son iguales para los diferentes periodos de planificación. Bajo estas circunstancias, cada instancia puede representarse como una instancia del problema presentado en Ibarra-Rojas *et al* (2012) donde el primer y último viaje de cada periodo de planificación están restringidos al intervalo dado por (3) y (4) en DHSM. Entonces, podemos usar una de las formulaciones reforzadas de DHS propuestas en Ibarra-Rojas *et al* (2012) e implementar CPLEX para hacer una comparación contra el algoritmo MVNS para resolver DHS. La Tabla 2 muestra la desviación relativa promedio (gap) entre la solución factible encontrada con MVNS y el valor de la cota superior encontrada con CPLEX, el tiempo promedio de ejecución en segundos (tiempo) y las respectivas desviaciones estándar ( $g\_stdev$  y  $t\_stdev$ ). Cabe mencionar, que el tiempo de ejecución para las instancias de tipo A1 mediante el enfoque exacto es de menos de 1 minutos mientras que para las instancias de A2-A4 está limitado a una hora de tiempo de ejecución.



TABLA 2: Gap y Tiempo de Implementar MVNS

Tipo instancia	A1	A2	A3	A4
gap	4.65%	8.72%	9.58%	11.64%
g_stdev	4.09	3.45	3.05	2.38
tiempo	3.31 s	7.56 s	10.53 s	15.72 s
t_stdev	2.38	4.21	4.49	5.76

Dado que el gap encontrado por CPLEX es muy pequeño para todas las instancias, podemos hacernos una idea bastante precisa de la calidad de MVNS. Podemos notar que en instancias pequeñas tenemos soluciones con una desviación promedio menor a 5% respecto a la cota. Sin embargo, mientras el tamaño de las instancias crece, la eficiencia de nuestro algoritmo disminuye. Aún para instancias de 200 líneas y 20 nodos de sincronización, podemos ver que la calidad de la solución es razonablemente buena mientras que el tiempo de cómputo es lo suficientemente pequeño para una implementación en un caso real donde pudiera ser necesario recalcular horarios debido a cambios imprevistos en el sistema. En resumen, nuestro enfoque de solución es el único capaz de encontrar soluciones con menos de 12% de desviación relativa respecto al óptimo para instancias grandes de DHSM en tiempos de cómputo razonables.

## 5.2 Impacto del Enfoque Multiperiodo

La principal ventaja de nuestro enfoque, es que evitamos el supuesto de que el sistema de transporte puede ser modelado de forma determinista en un mismo periodo de planificación para todas las líneas. Más aún, en los casos donde los tiempos de arribo a los nodos de sincronización son comparables con la longitud del periodo de planificación, las sincronizaciones entre viajes pertenecientes a periodos diferentes se vuelven necesarias puesto que muchas sincronizaciones serían ignoradas por el enfoque de un solo periodo. Para ejemplificar estos casos numéricamente, generamos un nuevo tipo de instancias llamadas A5 con 50 líneas y 5 nodos de sincronización considerando 10 periodos de planificación de 60 minutos y  $f_s^i \in [6,8]$ ;  $\frac{\Delta_i}{\mu_i} 100\% \in [15\%, 25\%]$ ; tiempos de viaje cada pareja de líneas  $(i, j \in J(i))$  y nodo  $b$  tal que  $t_{pb}^i \in [40,60]$  y  $t_{qb}^j \in [15,25]$ ; número de parejas de líneas a sincronizar en cada nodo entre  $[1,7]$ . Generamos 10 instancias para comparar el número de sincronizaciones alcanzadas al unir horarios obtenidos de la formulación de Ibarra-Rojas *et al* (2012) para un solo periodo (un periodo) y las alcanzadas al resolver nuestra formulación DHSM usando MVNS (multiperiodo). La Figura 5 muestra dicha comparación donde el eje horizontal representa las instancias, el eje vertical representa el número de sincronizaciones y los porcentajes debajo del eje horizontal representan la mejora alcanzada usando nuestra propuesta de solución.

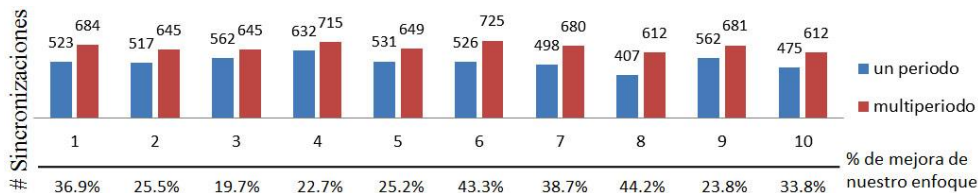


FIGURA 5: Sincronizaciones Logradas por Unir Horarios de un Solo Periodo y de Implementar el MVNS para Resolver el DHSM

Los resultados numéricos muestran que incluso sin garantizar optimalidad, el enfoque multiperiodo permite modelar de una forma más general el problema de horarios con sincronización y nuestra propuesta de solución es lo suficientemente eficiente para encontrar soluciones de buena calidad de instancias grandes.

## 6. CONCLUSIONES

La sincronización de líneas de transporte es un proceso complicado, aunque tiene beneficios importantes como favorecer la transferencia de pasajeros y disminución de congestión de tráfico para líneas que comparten segmentos de rutas.

Nuestro problema de diseño de horarios con sincronización en múltiples periodos (DHSM) maximiza el número de eventos que permiten una transferencia adecuada de pasajeros y eventos que evitan saturación de autobuses de diferentes líneas en paraderos comunes. Más aún DHSM permite la sincronización entre viajes que pertenecen a diferentes periodos de planeación específicos para cada línea. Esta característica permite modelar casos mucho más generales comparados con el enfoque de un solo periodo de planeación, pero también lo hace más intratable para solvers comerciales.

Cabe mencionar que nuestro objetivo puede estar en conflicto con la minimización del tiempo que pasan los usuarios en todo el sistema. Por esta razón, es de vital importancia no perturbar demasiado la regularidad de servicio a fin de no causar una mala imagen.

Debido a la intratabilidad de DHSM, implementamos una Búsqueda de Vecindario Variable multiarreglo (MVNS por sus siglas en inglés) para resolver el problema. Esta MVNS utiliza la estructura del problema para definir la región factible y explorarla de forma eficiente. La implementación de este tipo de algoritmos es la única herramienta disponible hasta el momento para encontrar, en un tiempo de cómputo razonable, soluciones factibles de buena calidad que mejoran desde 19.7% hasta 44.2% el enfoque de unir horarios obtenidos resolviendo óptimamente formulaciones basadas en un solo periodo de planificación. Sin embargo, la definición de los operadores, vecindarios y búsquedas locales, abre la puerta al desarrollo de otras metaheurísticas para mejorar los resultados obtenidos en este estudio e implementarlas en problemas similares, siendo también necesario, el cálculo de cotas superiores para el problema DHSM a fin de medir la calidad de las soluciones obtenidas por MVNS de forma más eficiente.

## AGRADECIMIENTOS

Ibarra-Rojas a la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile y a CONICYT/FONDECYT proyecto No. 3140358 por el financiamiento de su postdoctorado. También se agradece al Centro Desarrollo Urbano Sustentable (CEDEUS), Conicyt/Fondap/15110020. Finalmente, agradecemos a los revisores que ayudaron a mejorar la calidad de este artículo.

## REFERENCIAS

- Bessiere C. (2006) **Constraint propagation**. Reporte Técnico LIRMM 06020, University of Montpellier.
- Bookbinder, J.H. y A. Désilets (1992) Transfer optimization in a transit network, **Transportation Science** 26, 106-118.
- Ceder, A. (2001) Efficient timetabling and vehicle scheduling for public transport, en: Voß, S., Daduna, J.R. (Eds.), **Computer-Aided Scheduling of Public Transport**, Springer Berlin Heidelberg. Volume 505 of Lecture Notes in Economics and Mathematical Systems, pp. 37-52.
- Ceder, A., B. Golany y O. Tal (2001) Creating bus timetables with maximal synchronization, **Transportation Research A**, 35, 913-928.
- Ceder, A. y O. Tal (2001) Designing synchronization into bus timetables, **Transportation Research Record**, 1760, 28-33.
- Cevallos, F. y F. Zhao (2006) Minimizing transfer times in public transit network with genetic algorithm, **Transportation Research Record**, 1971, 74-79.
- Chakroborty, P., K. Deb y P.S. Subrahmanyam (1995). Optimal scheduling of urban transit systems using genetic algorithms, **Journal of Transportation Engineering**, 121, 544-553.
- Deb, K. y P. Chakroborty, (1998) Time scheduling of transit systems with transfer considerations using genetic algorithms. **Evolutionary Computation**, 6, 1-24.
- Desaulniers, G. y M. Hickman (2007) Public transit, en: Barnhart, C., Laporte, G. (Eds.), **Handbooks in operations research and management science**, North-Holland. Volume 14 of Transportation, 69-128.
- Eranki, A. (2004). **A Model to Create Bus Timetables to Attain Maximum Synchronization Considering waiting times at Transfer Stops**. Tesis maegister, Department of Industrial and Management Systems Engineering, University of South Florida, EUA.
- Guihaire, V. y J.K. Hao (2010). Transit network timetabling and vehicle assignment for regulating authorities. **Computers and Industrial Engineering**, 59(1), 16-23.
- Hansen, P. y N. Mladenović (2003) Variable neighborhood search, **Handbook of metaheuristics**, 145-184.
- van den Heuvel, A., J. van den Akker y M. van Kooten (2008). **Integrating timetabling and vehicle scheduling in public bus transportation**. Reporte Técnico UU-CS-2008-003, Department of Information and Computing Sciences, Utrecht University, Holanda.
- Ibarra-Rojas, O.J. (2013). **Models and Algorithms for Transit Network Planning**. Tesis PhD, Posgrado en Ingeniería de Sistemas, Universidad Autónoma de Nuevo León, México.
- Ibarra-Rojas, O.J., P. Fouilhoux, S. Kedad-Sidhoum y Y.A. Rios-Solis (2012). **Valid inequalities for the synchronization of bus timetabling**. Reporte Técnico PISIS-2012-02, Posgrado en Ingeniería de Sistemas, Universidad Autónoma de Nuevo León, México.
- Ibarra-Rojas, O.J. y Y.A. Rios-Solis (2012). Synchronization of bus timetabling, **Transportation Research B**, 46, 599-614.
- Jansen, L.N. y O.A. Nielsen (2002). Minimizing passenger transfer times in public transport timetables, en: **7th Conference of the Hong Kong Society for Transportation Studies, Transportation in the information age**, Hong Kong. pp. 229-239.
- Kwan, C.M. y C.S. Chang (2008). Timetable synchronization of mass rapid transit system using multiobjective evolutionary approach. **IEEE Transactions on Systems, Man, and Cybernetics, Part C**, 38, 636-648.
- Schröder, M. e I. Solchenbach (2006). **Optimization of Transfer Quality in Regional Public Transit**. Reporte Técnico 84. Berichte des Fraunhofer Instituts for Techno-und-Wirtschafts.
- Wong, R.C.W., T.W.Y. Yuen, K.W. Fung y J.M.Y. Leung (2008). Optimizing timetable synchronization for rail mass transit, **Transportation Science**, 42, 57-69.