



Mecánica Computacional Vol XXXI, págs. 3523-3530 (artículo completo)  
Alberto Cardona, Paul H. Kohan, Ricardo D. Quinteros, Mario A. Storti (Eds.)  
Salta, Argentina, 13-16 Noviembre 2012

## UNA ESTRATEGIA PARALELA CON SIMULATED ANNEALING PARA EL PROBLEMA DEL TRANSPORTE PÚBLICO INTERURBANO

Diego Alejandro Rodríguez<sup>a,b,c</sup>, Ana Carolina Olivera<sup>c</sup> y Néilda Beatriz Brignole<sup>b,c</sup>

<sup>a</sup>*Facultad de Ciencias Exactas, Universidad Nacional de Salta (UNSa),  
Av. Bolivia 5150, Salta, Argentina*

<sup>b</sup>*Planta Piloto de Ingeniería Química (PLAPIQUI) Complejo CCT-UAT, CONICET Camino la  
Carrindanga Km. 7, 8000 Bahía Blanca, Argentina*

<sup>c</sup>*Laboratorio de Investigación y Desarrollo en Computación Científica (LIDeCC), Departamento de  
Ciencias e Ingeniería de la Computación (DCIC), Universidad Nacional del Sur (UNS), Av. Alem  
1253, Bahía Blanca, Blanca, Argentina,*

*drodriguez@plapiqui.edu.ar, aco@cs.uns.edu.ar, dybrigno@criba.edu.ar*

**Palabras claves:** procesamiento paralelo, Simulated Annealing, MALLBA, transporte público interurbano, optimización.

**Resumen.** En este artículo se presenta un enfoque paralelo de Simulated Annealing para la resolución del problema de transporte público interurbano. El objetivo del estudio es optimizar el recorrido y las frecuencias de los autobuses pertenecientes a una línea de transporte, de manera que se minimice el costo por la prestación del servicio y se obtengan soluciones en un tiempo conveniente. Para ello, se implementó un algoritmo paralelo que utiliza la estructura de la librería MALLBA. La evaluación de las potenciales soluciones se lleva a cabo a través del software de simulación SUMO. Considerando el tiempo de cómputo que insumen las evaluaciones de las potenciales soluciones, se propone una versión paralela que disminuye dichos tiempos sin afectar la calidad de las soluciones obtenidas. El rendimiento del algoritmo paralelo es analizado en una línea de transporte interurbano real que une las ciudades de Bahía Blanca y Punta Alta en Argentina.

## 1 INTRODUCCION

El transporte público representa uno de los factores más importantes para el éxito de una región, afectando directamente la calidad de vida de sus habitantes y visitantes. Por distintas razones, muchas personas viajan todos los días de una ciudad a otra teniendo que enfrentar problemas como la congestión del tránsito, la escasez de lugares para estacionar o el incumplimiento de los horarios del servicio público. Reconociendo que son muchos los factores a tener en cuenta, una cuidadosa planificación del sistema de transporte público resulta indispensable.

La principal causa al problema es la creciente demanda, y un usual intento de solución es el incremento de los recursos. Aumentar y reorganizar la flota de vehículos y el personal de servicio es una alternativa sobre la que hay muchos estudios y resultados (Wren, 1999). Sin embargo, cuando es imposible incorporar nuevos autobuses y choferes a la línea, una política razonable consiste en la optimización de las rutas y frecuencias de autobuses con el fin de mejorar significativamente el sistema de transporte interurbano (Baaj y Mahmassani, 1991). Esta posibilidad está regulada por las autoridades y afecta a la asignación de tareas (Ceder y Wilson, 1986).

A pesar de que es similar al transporte público urbano, el transporte público interurbano posee algunas diferencias importantes que requieren un tratamiento especial: a) por lo general tiene menos paradas y están más alejadas entre ellas, b) hay menos interacción entre vehículos, provocando un menor número de detenciones no planificadas, c) existen tramos de ruta con características especiales (por ejemplo, mayores límites de velocidad, o más de dos carriles). La mayor parte de las investigaciones sobre optimización de rutas está relacionada al transporte urbano, en lugar de concentrarse en el transporte interurbano (Guihaire y Hao, 2008, Su y Chang, 2010, Yan et al., 2006, Yan y Chen, 2002, Olivera et al., 2009). En este grupo hay sólo unos pocos artículos que consideran a la demanda un factor estocástico (Yan et al., 2006, Olivera et al., 2009).

Entre algunos de los estudios sobre el tráfico urbano, Salzborn (1980) modela el diseño de la frecuencia con el fin de minimizar la flota y los tiempos de espera para los usuarios. Van Nes (1988) propone evaluar todas las rutas a partir de una frecuencia cero e ir aumentándola gradualmente de acuerdo a la eficiencia marginal obtenida. Sin embargo, estudios más recientes (Guihaire y Hao, 2008) han concluido que es apropiado abordar el problema de las frecuencias y rutas de manera simultánea.

En Rodríguez et al. (2011) un algoritmo Simulated Annealing (SA) se comparó con un Algoritmo Genético (GA), ambos implementados en la arquitectura MALLBA (Alba et al., 2007). La versión paralela del método descrito en Rodríguez et al. (2011) se presenta en este trabajo. Notables mejoras fueron alcanzadas gracias al aprovechamiento de las ventajas que ofrecen las nuevas tecnologías en términos de potencia y velocidad cálculo. La implementación de una estrategia paralela en nuestro modelo surge de la necesidad de obtener soluciones de buena calidad reduciendo los tiempo de computo (Crainic y Toulouse, 1998). En investigaciones sobre transporte se destaca la publicación de Florian y Gendreau (2001), la cual describe las importantes contribuciones del paralelismo en metaheurísticas aplicadas a los modelos y algoritmos empleados en la planificación del transporte y la simulación.

Analizamos el rendimiento de nuestro Simulated Annealing Paralelo (SAP) con el objetivo de mejorar una línea interurbana real que une Bahía Blanca y Punta Alta, teniendo en cuenta el flujo de tráfico real entre ambas ciudades de Argentina. Además, se realizara una comparación entre nuestro trabajo anterior sobre el algoritmo Simulated Annealing Secuencial

(SAS) y esta nueva versión paralela.

El resto del trabajo se organiza de la siguiente manera: la formulación del problema se presenta en la sección 2. En la sección 3 mostramos los detalles de la estrategia paralela para nuestro algoritmo. El análisis experimental y los resultados obtenidos se discuten en la sección 4. Finalmente, en la sección 5 formulamos las conclusiones del presente trabajo.

## 2 FORMULACION DEL PROBLEMA

La entidad administradora del servicio debe asegurar la accesibilidad al transporte público de la mejor manera, además de intentar obtener por ello el máximo beneficio posible. Nuestro trabajo puede entenderse como un módulo que pertenece a un sistema mucho más complejo de modelos de decisión. Consideramos que un buen diseño de rutas y una adecuada frecuencia de salidas para los autobuses tienen un impacto directo en los resultados de la gestión.

Se describe una línea interurbana como una secuencia de paradas, que deberán ser visitadas por los autobuses con una frecuencia determinada. Las 22 paradas existentes están distribuidas en las ciudades de Bahía Blanca y Punta Alta conociendo el punto inicial y final del recorrido que tiene una longitud aproximada de 30 km. La frecuencia de partida de los autobuses, por disposiciones gubernamentales, debe estar entre los 20 y 50 minutos y el tiempo de simulación utilizado fue de 8 horas (una jornada completa).

El objetivo del modelo es minimizar el costo total de brindar un servicio que satisfaga las condiciones mencionadas. Así, para cada configuración se tendrá en cuenta el número total de kilómetros recorridos por cada uno de los autobuses, los tiempos demandados y la cantidad de detenciones no planificadas que ocurrieron, penalizando aquellos autobuses que no completaron el recorrido.

Una solución a este problema es representado como un vector de números enteros que identifican a cada una de las paradas existentes, junto a un valor, también entero, que indica la frecuencia de partida para los autobuses. Además, mantenemos un vector similar que conserva la información para identificar a que ciudad pertenece cada una de las paradas (Rodríguez et al. 2011). Un ejemplo de la representación de una solución para el IPTP se puede ver en la Figura 1.

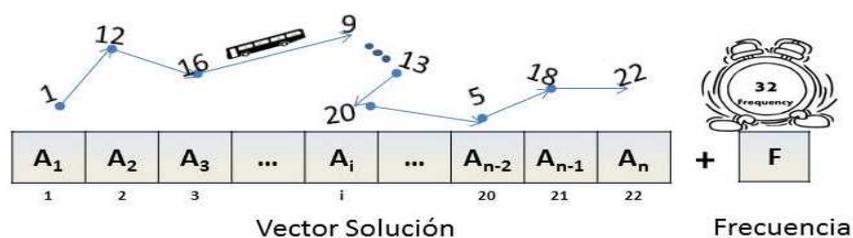


Figura 1: Representación de la solución.

Para evaluar cada una de las potenciales soluciones se utiliza una función de aptitud (ver Ecuación 1). Los valores de la Ecuación 1 para las soluciones generadas por SAP se obtienen a través del software de simulación SUMO (Behrisch et al., 2011). De esta manera podemos definir propiedades tales como la aceleración y desaceleración de los vehículos, la dirección de las calles, los tiempos de espera, entre otros. SUMO permite obtener simulaciones precisas sobre la conducta del tráfico, además, es de código abierto, altamente portable y soporta mapas completos de ciudades reales obtenidos desde GoogleMaps u OpenStreetMap (Rodríguez et al., 2011).

$$\text{Min } Z = \frac{(T + R + ST)}{V} \quad (1)$$

### Símbolos

$t_i$  El tiempo total del viaje para el autobús  $i$  in segundos

$T = \sum_{i=0}^{NV+V} t_i$

$v_i$  El autobús  $i$ .

$V$  El número de autobuses que llegan a destino.

$NV$  El número de autobuses que no llegan a destino.

$r_i$  El costo de la ruta para el autobús  $i$ .

$R = \sum_{i=0}^{NV+V} r_i$

$s_i$  El número de veces que el autobús  $i$  debe detenerse sin haber planificado.

$ST = \sum_{i=0}^{NV+V} s_i$

$S$  El tiempo total de simulación.

El procedimiento general del algoritmo SA se puede resumir en:

1. Generar una solución potencial (SP) inicial. En nuestro caso será una solución aleatoria. Esto es completar el vector de 22 elementos enteros al azar.

2. Calcular la función objetivo para la configuración actual, almacenando aquella que ofrezca el menor costo durante la ejecución del proceso. Para el cálculo de dicha función utilizamos el simulador SUMO.

3. Realizar un proceso de generación de nuevas soluciones (NS) que nos permita explorar el espacio de búsqueda. En nuestro caso utilizamos un mecanismo de selección de vecinos que elige aleatoriamente dos paradas y procede a intercambiarlas de lugar, favoreciendo aquellas paradas que pertenecen a ciudades distintas.

4. La aceptación o no de nuevas soluciones dependerá de la diferencia de calidad (el valor del fitness) entre ambas soluciones y de la temperatura que actúa como parámetro de control en la metaheurística SA y que tiene como valor inicial una tasa de aceptación de soluciones propuestas mayor o igual al 99%. Dicho valor irá disminuyendo lentamente cada 10 iteraciones del algoritmo. Esto nos permitirá realizar una exploración del espacio de búsqueda en las primeras iteraciones y una explotación en las últimas dificultando la aceptación de soluciones peores. Así, se evita quedar atrapado en un óptimo local y se cumple la filosofía habitual de búsqueda de diversificar al principio e intensificar al final. Para mayores detalles sobre la filosofía del algoritmo ver Kirkpatrick et al. (1983).

Nuestro algoritmo (Ver Figura 2) incluye la generación de un archivo XML que contiene todos los datos necesarios para calcular la función de fitness, este archivo es enviado al simulador, SUMO procesa estos archivos, y devuelve el valor de fitness (o aptitud) que indica la calidad de la solución.

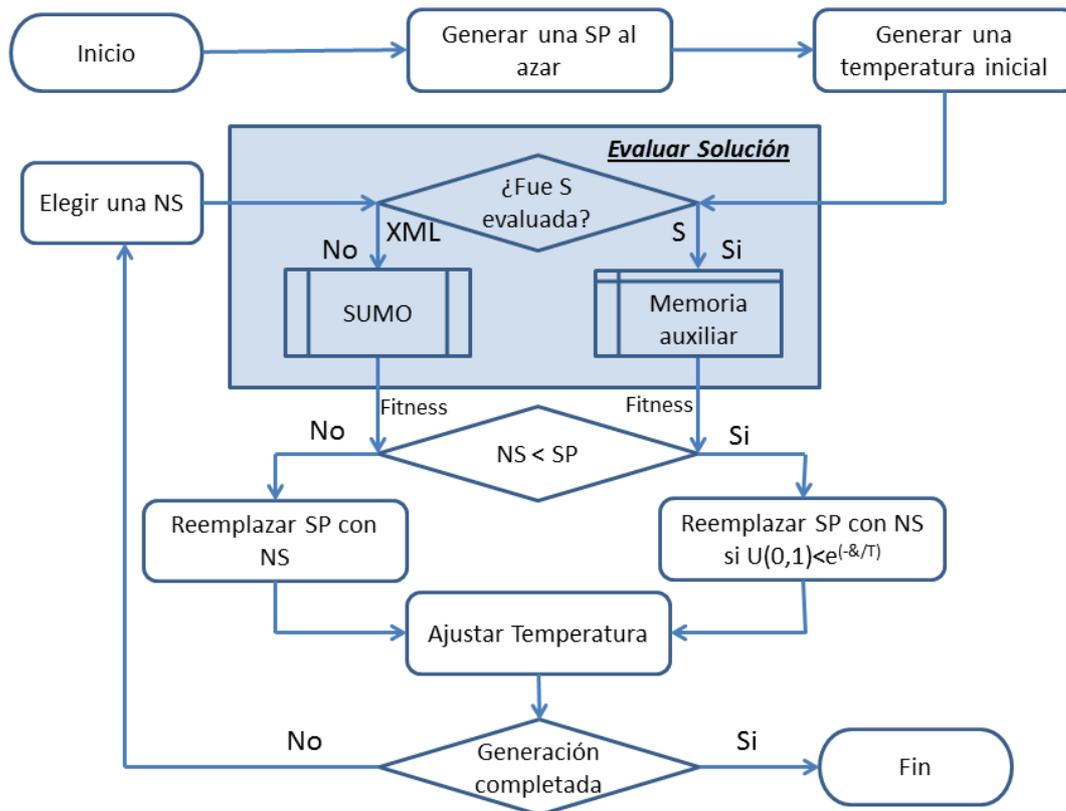


Figura 2: Representación del proceso SA

### 3 ESTRATEGIA PARALELA

Los algoritmos secuenciales suelen tornarse ineficientes al abordar problemas complejos, sobre todo cuando la evaluación de la función de fitness es muy costosa en cuanto a tiempo de cómputo. Además en un algoritmo secuencial, cuando se necesita mantener un conjunto de soluciones se necesita más memoria de la habitual y por lo general también ocurre que convergen prematuramente hacia valores sub-óptimos.

Entonces, para contrarrestar estos problemas surge la necesidad de usar técnicas que paralelicen la ejecución de los algoritmos. Un algoritmo que utiliza más de un procesador para llevar a cabo su ejecución, es llamado algoritmo paralelo. Esta característica permite ampliar la cantidad de sectores de búsqueda dentro del espacio de soluciones aumentando la diversidad y por ende minimizando la probabilidad de converger prematuramente.

La arquitectura de computación paralela que empleamos, siguiendo la arquitectura basada en esqueletos de la biblioteca MALLBA, permite la ejecución de algoritmos concurrentes en procesadores multi-núcleo. Un procesador de esta característica permite la ejecución de varias tareas en forma simultánea. Además, tiene la ventaja de que los núcleos al estar integrados a un mismo circuito obteniendo una reducción de los tiempos de comunicación entre los procesos, sin embargo estamos limitados por la cantidad de núcleos sin la posibilidad de poder expandirnos como cuando se trabaja con un cluster de computadoras.

Utilizamos una topología que distingue dos tipos de procesos, un proceso maestro (P0) que será el encargado de recolectar los resultados obtenidos en cada generación por cada uno de

los restantes procesos ( $P_i$ ), que serán 3 en el presente trabajo, debiendo mantener un estado global del algoritmo, (ver Figura 3). Utilizamos un modelo síncrono, lo que significa que el procesos maestro espera a que todos los procesos finalicen su tarea antes de iniciar la siguiente generación. La comunicación entre los procesos se realiza mediante el software Netstream (Alba, 2001), que nos ofrece una interfaz orientada a objetos sobre la biblioteca de pasos de mensajes MPI (Gropp et. al, 1996).

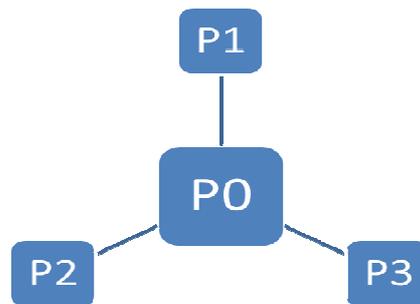


Figura 3: Topología Maestro-Esclavo

#### 4 RESULTADOS Y COMPARACIONES

La ejecución de nuestro algoritmo SAP se llevó a cabo utilizando parámetros con los mismos valores que para el SAS como se presenta en la Tabla 1. El equipo de cómputo utilizado fue una computadora IBM con procesador AMD Phenom QUAD CORE SERIE 9000 y 8GB de memoria RAM.

Evaluaciones	Longitud de la cadena de Markov	Temperatura Inicial
300	10	0.99

Tabla 1: Valores de los parámetros del algoritmo.

Los resultados del algoritmo paralelo propuesto y su par secuencial se resumen y se comparan en la Tabla 2, luego de 30 ejecuciones independientes. Para cada algoritmo se presentan los promedios y mejores valores de la función objetivo, el número de iteración en la cual se obtuvo ese mejor valor, el tiempo promedio de ejecución y las medidas para evaluar el rendimiento como lo son el Speedup y la eficiencia.

El Speedup es la razón entre el tiempo medio de ejecución de la versión secuencial y paralelo del mismo algoritmo. La eficiencia computacional es el cociente entre el Speedup y la cantidad de procesadores utilizados. Un valor de eficiencia mayor a 1 significa que los elementos de procesamiento en conjunto logran mejores resultados que por separado.

Algoritmo	# Procesadores	Mejor Fitness	Fitness Promedio	Iteración	Tiempo	Speedup	Eficiencia
SA Secuencial	1	92468.2	86770.7	289	12.27E+5	1.28	32%
SA Paralelo	4	91741.9	88842.6	263	95.53E+4		

Tabla 2: Comparación del algoritmo secuencial con el paralelo.

De los resultados obtenidos podemos afirmar que no existen diferencias significativas entre los valores promedios de fitness, siendo levemente mejor el algoritmo secuencial. Sin embargo al analizar los tiempos de cálculo, podemos observar una marcada diferencia en favor del algoritmo paralelo. La utilización de cuatro procesadores otorga una ganancia en la eficiencia del procesamiento cercano al 32%. Con respecto al Speedup, este escenario muestra un comportamiento sub-lineal. A pesar de que, naturalmente, la mejora no es proporcional al número de procesadores, este comportamiento lo consideramos satisfactorio y prometedor.

También podemos observar en la Figura 4 que el modelo paralelo encuentra mejores valores en las primeras iteraciones y no necesita tantas iteraciones para encontrar una buena solución final. Motivo por el cual permite concluir que la ganancia de tiempo puede ser aún mayor comparado con el algoritmo secuencial.

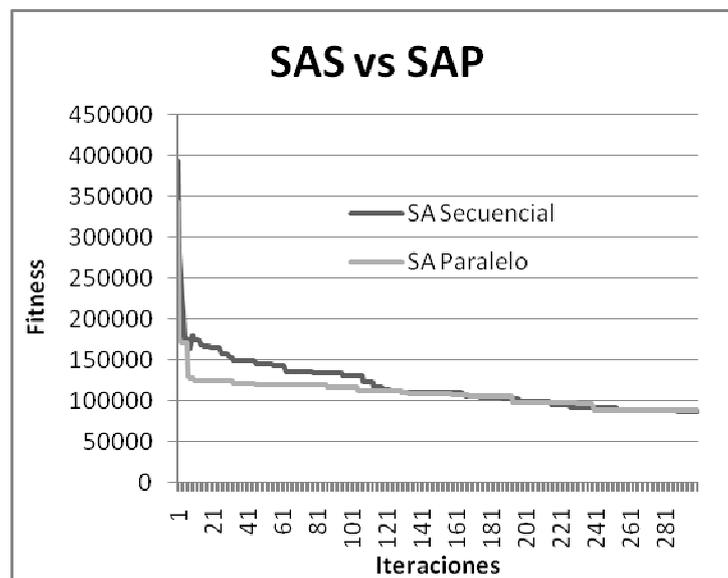


Figura 3: Evolución del algoritmo Paralelo y Secuencial.

## 5 CONCLUSIONES

Un mejor transporte no sólo se logra con mejor infraestructura. También resulta muy importante optimizar el uso de los recursos. Como solución al problema del transporte público interurbano, proponemos un algoritmo Simulated Annealing Paralelo que considera como función objetivo la minimización de los costos de funcionamiento de una línea de transporte existente entre las ciudades de Bahía Blanca y Punta Alta.

El concepto de paralelismo surge con la necesidad de acelerar los tiempos de cómputo de cualquier proceso. Mediante la paralelización del algoritmo SA, se consigue reducir el tiempo de ejecución sin afectar la calidad de las soluciones obtenidas, esto dicho en comparación con nuestro antecesor: el algoritmo SA secuencial.

La paralelización del algoritmo según un modelo de maestro-esclavo también contribuye a la diversidad para cada proceso en ejecución. Debemos recalcar que en el problema del transporte interurbano se trabaja con instancias más grandes a la presentada en este trabajo, implicando un elevado grado de procesamiento y conviniendo paralelizar para reducir la ejecución del algoritmo. Las múltiples aplicaciones de estas técnicas de optimización sugieren que es posible aplicarlos con éxito a distintos problemas de transporte.

## REFERENCIAS

- Alba, E., "Netstream: A Flexible and Simple Message Passing Service for LAN/WAN Utilization", E.T.S.I.I. Málaga. Dpto. de Lenguajes y Ciencias de la Computación, 2001.
- Alba E., Luque G., García-Nieto J.M., Ordonez G., and Leguizamón, G., MALLBA: A Software Library to Design Efficient Optimisation Algorithms Int. J. of Innovative Computing and Applications 2007 (IJICA), 1, 74-85, 2007.
- Baaj, M. H. and Mahmassani, H. S., An AI-Based Approach for Transit Route System Planning and Design. Journal of Advanced Transportation, Vol 25(2), 187-210, 1991.
- Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D., 2011. Sumo – simulation of urban mobility: An overview, in: SIMUL 2011, The Third International Conference on Advances in System Simulation, Barcelona, Spain, pp. 63–68, 2011.
- Ceder, A. and Wilson, N. H. M., Bus Network Design. Transportation Research, Vol 20B(4), 331-344, 1986.
- Crainic T., Toulouse M., Parallel Strategies for Meta-Heuristics. *Handbook of Metaheuristics*. Glover and Kochenberger editors. International Series in Operations Research & Management Science. Springer New York. Pp. 475-513, Vol. 57, 2003.
- Florian M. and Gendreau M.. Applications of parallel computing in transportation, Parallel Computing, Volume 27, Issue 12, pp. 1521-1522, 2001.
- Gropp W., Lusk E., Doss N., and Skjellum A.. A high-performance, portable implementation of the MPI message passing interface standard. Parallel Computing, 22:789:828, 1996.
- Guihaire V., Hao J. Transit Network Design and Scheduling: A Global Review. Transportation Research Part A: Policy and Practice, 42(10):1251-1273, 2008.
- Kirkpatrick, S., C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," Science 220, 671-680, 1983.
- Olivera A., Frutos M, Carballido J, Ponzoni I, Brignole N., Bus Network Scheduling Problem: Grasp + eas With Pisa \* simulation. In Bio-Inspired Systems: Computational and Ambient Intelligence, volume 5517 of *Lecture Notes in Computer Science*, pp. 1272-1279, 2009.
- Rodríguez D, Olivera A.C., Brignole N. B., Meta-heuristics applied to the intercity transport problem. ENIEF 2011 XIX Congreso sobre Métodos Numéricos y sus Aplicaciones en Rosario, Santa Fé, Argentina. Mecánica Computacional Vol XXX, pp 2169-2179, 2011.
- Su J.M. and Chang C.H., The Multimodal Trip Planning System of Intercity Transportation in Taiwan. Expert Systems with Applications, 37(10):6850-6861, 2010.
- Wren, A., Heuristics Ancient and Modern; Transport Scheduling Through the Ages. Leeds Artificial Intelligence Seminar Series, University of Leeds, 1999.
- Van Nes, R., Hamerslag, R., Immers, B.H., Design of Public Transport Networks. Transportation Research Record 1202, 74-83, 1988.
- Yan S., Chen H., A Scheduling Model and a Solution Algorithm for Inter-city Bus Carriers. Transportation Research Part A: Policy and Practice, 36(9):805-825, 2002.
- Yan S., Chi C., Tang C., Inter-city Bus Routing and Timetable Setting Under Stochastic Demands. Transportation Research Part A: Policy and Practice, 40(7):572-586, 2006.