

# IMPLEMENTACIÓN DE ALGORITMOS DE BÚSQUEDA GLOBAL PARA LA CALIBRACIÓN DE ASIGNACIÓN DE TRANSPORTE PRIVADO

José Imbarack <sup>1</sup>, Javiera Olguín <sup>2\*</sup>, Roberto Cominetti <sup>1</sup>

<sup>1</sup> Universidad Adolfo Ibáñez, Chile

<sup>2</sup> Cityplanning, Chile

---

\*Autor para correspondencia:

Jolguin@cityplanning.cl

## RESUMEN

Este trabajo explora el uso de algoritmos de optimización para la calibración del submodelo de asignación de transporte privado de los modelos de transporte ESTRAUS y VIVALDI. Tradicionalmente se ha utilizado el algoritmo Hooke & Jeeves para estos fines. Este trabajo busca evaluar el uso de algoritmos alternativos que consideren los avances de la literatura sobre algoritmos de optimización y las capacidades de cómputo actuales.

Se consideran tres algoritmos de búsqueda global: Nelder-Mead, Simulated Annealing y Particle Swarm, analizados comparativamente con el método Hooke & Jeeves.

Para el análisis comparativo de los algoritmos se usaron las redes de modelación de tres ciudades chilenas. Se considera tanto las mejoras en la función objetivo, el tiempo de calibración y la consistencia de los parámetros.

Se confirma la tendencia del algoritmo Hooke & Jeeves a quedar atrapado en mínimos locales. En contraste, los algoritmos alternativos encuentran soluciones mejores, aunque con un mayor tiempo de ejecución. Si los algoritmos alternativos se restringen a una región de búsqueda apropiada, logran encontrar buenas soluciones en un tiempo competitivo. Se analiza el uso de estrategias combinadas de los algoritmos de búsqueda global. Se concluye que las todas las estrategias probadas logran mejorar la calidad de las calibraciones para las ciudades analizadas.

**Palabras clave:** optimización, asignación, ESTRAUS, VIVALDI

## ABSTRACT

*Optimization algorithms are routinely used in the process of parameter estimation in physical and/or engineering models. This work explores the use of optimization algorithms for the calibration of the private transport allocation sub-model of the ESTRAUS and VIVALDI transport models.*

*The Hooke and Jeeves algorithm has traditionally been used for this purpose. This project seeks to assess the use of alternative algorithms that consider the progress of the literature on optimization algorithms and the increased computing power currently available.*

*We consider three of global search algorithms: Nelder-Mead, Simulated Annealing and Particle Swarm, which were analyzed comparatively with Hooke & Jeeves.*

*For the comparative analysis, the modelling networks of three Chilean cities were used. The analysis considered both the improvements in the objective function, as well as the running time and the consistency of the parameters.*

*We confirmed the tendency of the Hooke & Jeeves algorithm to be trapped at local minima. In contrast, the three alternative algorithms find better solutions, although with longer execution times. When the alternative algorithms are restricted to an approximate search region, they manage to find good solutions in a competitive time. We propose the use of hybrid strategies that combine the global search algorithms. All the strategies tested improve the quality of the calibrations in the cities analyzed.*

**Keywords:** optimization, allocation, ESTRAUS, VIVALDI

---

---

## 1. INTRODUCCIÓN

La planificación del transporte urbano en Chile ha sido desde sus inicios un tema muy relevante para el Ministerio de Transporte y Telecomunicaciones (MTT). Este ministerio cuenta con un organismo técnico especializado llamado SECTRA (Programa de Vialidad y Transporte Urbano) que se encarga, dentro de otras tareas, del desarrollo de los planes de transporte de todas las grandes ciudades y ciudades de tamaño intermedio en Chile.

Los Planes Maestros de Transporte (o Planes del Sistema de Transporte Urbano, STUs) son en general realizados en dos fases; una primera fase de construcción y calibración del modelo de transporte, ya sea el modelo ESTRAUS para grandes ciudades o el modelo VIVALDI para ciudades intermedias, y una segunda fase donde se desarrolla el plan de transporte usando el modelo construido en la fase previa.

Tanto ESTRAUS como VIVALDI han sido desarrollados por el Estado de Chile y corresponden a modelos de 4 Etapas, es decir, consideran las etapas de Generación y Atracción de Viajes, Distribución, Partición Modal y Asignación. Mientras ESTRAUS resuelve las cuatro etapas de forma simultánea, VIVALDI lo hace de forma secuencial. La construcción y calibración del submodelo de asignación necesita, a su vez, del uso de un algoritmo de optimización externo a ESTRAUS y VIVALDI.

Para la calibración de la asignación de redes de transporte tradicionalmente se ha utilizado el algoritmo de búsqueda local Hooke & Jeeves (Hooke y Jeeves, 1961). Este algoritmo, desarrollado hace 60 años, se encuentra desactualizado respecto a los avances que han presentado tanto los algoritmos de optimización, particularmente los de búsqueda global, como los grandes avances en las capacidades de cómputo que permiten hoy en día cálculos mucho más complejos que hace 6 décadas.

En el presente trabajo se desarrolla una herramienta que mejora la calidad de la calibración del submodelo de asignación de ESTRAUS y VIVALDI, con respecto al algoritmo de Hooke & Jeeves. Para la obtención de los resultados se aplica un análisis de tres etapas, donde las dos primeras permiten definir las estrategias a utilizar en la tercera etapa del análisis.

## 2. ALGORITMOS DE OPTIMIZACIÓN

Existe una amplia variedad de algoritmos para problemas de optimización, los cuales pueden ser clasificados según sus características y formas de resolución. Dentro de estas clasificaciones destacan dos mundos que agrupan gran parte estos métodos: los algoritmos de búsqueda local y los de búsqueda global. Los primeros, se caracterizan por encontrar soluciones a partir de la búsqueda en el vecindario del punto inicial a través de métodos simples de iteración o diferentes heurísticas. Para problemas de optimización no convexa estos métodos suelen quedar atrapados en óptimos locales. Por su parte, los algoritmos de búsqueda global se caracterizan por tener un rango de búsqueda mayor a los primeros, es decir, utilizan metodologías más complejas para no solo buscar soluciones cercanas al punto inicial y extender la búsqueda a todo el espacio de soluciones factibles. Si bien los métodos globales mejoran significativamente la calidad de las soluciones, en general presentan un mayor tiempo de ejecución y tampoco garantizan encontrar un óptimo global. La incorporación del tiempo de ejecución de los algoritmos se ha considerado relevante dado el

contexto en el cual se pretende usar este tipo de modelos, esto es, la calibración de redes de transporte urbano dentro de estudios SECTRA que poseen tiempos acotados de desarrollo. Sin embargo, esta consideración pudiese no ser relevante si los plazos de desarrollo establecidos son holgados.

## 2.1 Algoritmo de Hooke & Jeeves

Tradicionalmente, la calibración de parámetros del módulo de asignación de ESTRAUS se ha realizado usando el algoritmo de Hooke & Jeeves (Hooke y Jeeves, 1961). Este algoritmo con características de búsqueda local es el único recomendado por SECTRA para realizar este proceso según la metodología de análisis del sistema de transporte en ciudades medianas y grandes (SECTRA, 2008). Se trata de un método iterativo para problemas de optimización con función objetivo no convexa y sin una expresión explícita para sus derivadas. El algoritmo itera de forma alternada las dos etapas siguientes, a partir de un punto inicial dado:

1. *Búsqueda exploratoria.* Esta etapa determina una dirección local de descenso a partir del punto actual. El procedimiento consiste en modificar secuencialmente cada uno de los parámetros a calibrar sumando en primera instancia una cantidad  $\delta$  y evaluando la función objetivo en el punto modificado. Si la función objetivo decrece se acepta el cambio, de lo contrario en segunda instancia se resta la misma cantidad  $\delta$  aceptándose el cambio con el mismo criterio. Si ninguno de estos cambios produce un descenso se mantiene el valor actual del parámetro. Este procedimiento se aplica secuencialmente a cada uno de los parámetros siguiendo un orden preestablecido. Si al término de este proceso no se logra disminuir la función objetivo, se reduce el valor de  $\delta$  y se repite la búsqueda exploratoria. Una vez que la búsqueda exploratoria es exitosa y logra un descenso de la función objetivo se procede con la siguiente etapa (2).
2. *Patrón de movimiento.* La búsqueda exploratoria entrega una dirección en la cual la función objetivo decrece. En esta segunda etapa se avanza a lo largo de esa dirección realizando varios pasos de tamaño  $\alpha$  mientras se observe una reducción de la función objetivo. Apenas se detecta un incremento de la función objetivo se toma el mejor punto alcanzado como nuevo punto base para volver a la etapa 1 a realizar una nueva búsqueda exploratoria.

El algoritmo se detiene apenas  $\delta$  cae por debajo de una tolerancia preestablecida  $\epsilon$ , reportándose el punto actual como solución del proceso de optimización.

Cada iteración requiere entre  $n$  y  $2n$  evaluaciones de la función objetivo para la búsqueda exploratoria, y otras tantas en la segunda etapa, siendo éste el principal costo computacional<sup>1</sup>.

Cabe notar además que el parámetro  $\delta$  solo puede decrecer a lo largo de las iteraciones, lo cual tiende a limitar la región del espacio de soluciones que se logra explorar. Debido a ello, y al igual que todos los métodos de búsqueda local, el algoritmo tiende a quedar atrapado en mínimos locales. Por consiguiente, para obtener una buena calibración se debe aplicar el algoritmo a partir de varios

---

<sup>1</sup> Donde  $n$  es el número de variables de la función objetivo, es decir, los parámetros a calibrar según caso.

---

puntos iniciales a fin de explorar distintas regiones del espacio factible, con el consiguiente aumento de los tiempos de cómputo.

## 2.2 Algoritmos alternativos

Dadas las limitantes señaladas del algoritmo de Hooke & Jeeves, resulta de interés su comparación con métodos de búsqueda global, así como con métodos locales alternativos que adaptan de forma dinámica la vecindad de búsqueda, ya sea reduciéndola o expandiéndola, en función del progreso observado del algoritmo. En ambos casos, estos métodos logran explorar mayores regiones del espacio factible en una misma corrida del algoritmo, sin necesidad de considerar múltiples puntos iniciales.

Cabe notar que la función objetivo del problema de calibración de parámetros del módulo de asignación, la cual mide la discrepancia entre los flujos observados y los flujos modelados, requiere resolver un modelo de equilibrio de transporte para cada conjunto de parámetros. Dicho equilibrio se obtiene mediante un software de tipo caja negra, lo cual solo permite obtener el valor de la función objetivo, pero no así sus derivadas respecto de los parámetros de calibración, y todo ello con un costo computacional significativo. Este hecho lleva a descartar el uso de métodos de tipo gradiente o Newton, dejando como única opción los métodos libres de derivadas, también conocidos como Derivative Free Optimization (DFO). Este tipo de algoritmo resulta apropiado para problemas en los cuales el cálculo de derivadas resulta demasiado costoso o imposible, como en el presente caso en que la función objetivo se obtiene mediante un proceso de simulación. Dentro de la familia DFO se escogieron tres métodos que han mostrado su eficacia en distintas áreas de la industria, a saber, los algoritmos de Nelder-Mead, Simulated Annealing, y Particle Swarm. Cabe mencionar que también se intentó un método de tipo gradiente basado en diferencias finitas, el cual mostró un pobre desempeño razón por la cual se desechó su uso.

- a) *Nelder-Mead*: Este algoritmo puede verse como un híbrido entre un método de búsqueda local y global, al usar una vecindad que se expande o contrae en función de la evolución del algoritmo. El método realiza un seguimiento a varios puntos en el espacio de soluciones los cuales forman un símplice (*i.e.* un polígono con  $n + 1$  vértices en dimensión  $n$ ). Dicho símplice se adapta de manera dinámica mediante 4 operaciones elementales: reflexión, expansión, contracción, y reducción. La idea central consiste en reemplazar iterativamente el vértice  $x$  con peor valor de la función objetivo por un nuevo punto  $x'$  obtenido mediante una reflexión respecto del baricentro del polígono. Si el punto reflejado  $x'$  es mejor que  $x$ , aunque no mejor que algún otro vértice, se sustituye  $x$  por  $x'$ . Si en cambio  $x'$  es mejor que todos los vértices anteriores, se prueba si conviene moverse más allá en la misma dirección, en cuyo caso tenemos una expansión. Si por el contrario  $x'$  resulta peor que  $x$ , se intenta reemplazar  $x$  por un punto intermedio entre  $x$  y el baricentro, y si este punto intermedio sigue siendo peor que  $x$  se encoge el símplice respecto de su baricentro. El algoritmo termina cuando todos los vértices entregan el mismo valor de la función objetivo dentro de una tolerancia predefinida. Para los detalles del método y su implementación se puede consultar Nelder y Mead (1965) y Gao y Han (2010). Según lo descrito, el símplice cambia de forma y se desplaza a través de la región factible permitiendo un mayor rango de exploración comparado con un método de búsqueda local puro. Notar que cada iteración

requiere a lo más 2 evaluaciones de la función objetivo, lo que se compara favorablemente respecto de Hooke & Jeeves que requiere al menos  $n$  evaluaciones por iteración.

- b) *Simulated Annealing*: Este algoritmo, también conocido como recocido simulado, consiste en una heurística de optimización global que emula el proceso de recocido termodinámico en metalurgia. El algoritmo utiliza un método probabilístico para encontrar el mínimo de una función  $f(x)$ , por lo general en dominios grandes. Cada iteración escoge un punto  $x'$  cercano al punto actual  $x$ , y decide si moverse hacia  $x'$  con una probabilidad dada por una ley de Gibbs. Mas exactamente, si  $f(x') \leq f(x)$  el sistema se mueve con certeza hacia  $x'$ , y en caso contrario se acepta el nuevo punto con probabilidad  $\exp[(f(x) - f(x'))/T]$ , donde  $T$  es un parámetro de temperatura que se hace decrecer lentamente hacia 0. El proceso se detiene cuando la temperatura es suficientemente pequeña o al alcanzar el límite de tiempo computacional disponible. Notar que el método puede saltar hacia estados de mayor energía, lo cual le permite escapar del entorno de un mínimo local y le confiere al método la capacidad de explorar otras regiones del espacio para concentrarse asintóticamente en torno a los estados de mínima energía global. Sin embargo, para que ello ocurra se requiere que la temperatura decaiga lentamente, lo cual depende de manera compleja tanto de la topografía de la función objetivo como del método de selección del vecino. En la práctica la velocidad de decaimiento de la temperatura se ajusta de manera empírica para cada problema en particular. Para mayores detalles del análisis e implementación del método ver Goffe et. al. (1994) y Goffe (1996).
- c) *Particle Swarm*: Nuevamente se trata de una heurística de optimización global la cual imita el comportamiento evolutivo de partículas o el movimiento de organismos vivos tales como una bandada de aves, un banco de peces, o un enjambre de abejas. El método considera una población de soluciones candidatas, o “*enjambre de partículas*”, las cuales se mueven por el espacio de búsqueda según reglas que consideran las posiciones y velocidades de las partículas. Los movimientos se ven influidos por la posición actual de cada partícula, así como por las mejores posiciones encontradas hasta el momento. En concreto, a partir de posiciones y velocidades iniciales para cada partícula, éstas se hacen evolucionar actualizando la posición de cada partícula según su velocidad actual, y modificando la velocidad de cada partícula mediante una combinación lineal entre la velocidad anterior y dos componentes que apuntan respectivamente en la dirección de la mejor posición visitada por la partícula y la mejor posición global observada por el enjambre hasta ese momento, cada una de ellas multiplicadas por un coeficiente aleatorio escogido uniforme en  $[0,1]$ . Para mayores detalles ver Kennedy y Eberhart (1995) o Zhan et al. (2009).

### 3. CALIBRACIÓN DE REDES DE TRANSPORTE

La calibración de modelos consiste en la búsqueda de los parámetros que permiten reproducir de la mejor forma un comportamiento observado en la realidad.

Para la calibración de redes de transporte se requiere una red de modelación previamente construida que represente razonablemente la ciudad, y conteos de flujos en arcos relevantes de la red. En el caso de redes de transporte privado, es necesario contar con conteos de flujo vehicular, idealmente clasificados según alguna tipología de vehículos predefinida. Dado que se suele definir más de un

período de modelación, es necesario calibrar cada período definido y, por lo tanto, se requiere disponer de conteos de flujos para cada uno de dichos períodos.

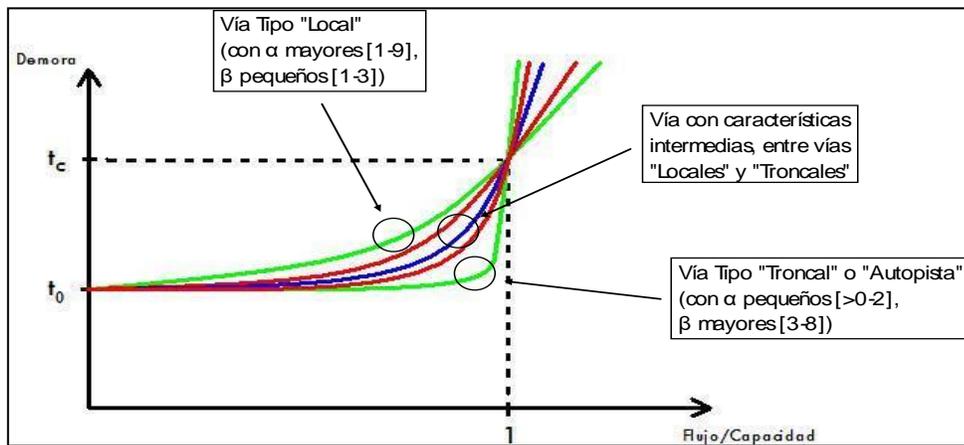
Por otro lado, para una correcta representación de los diferentes tipos de calles de una ciudad, se suelen definir categorías de arco, los cuales guardan cierta relación con aquellas categorías de vías definidas en los instrumentos de planificación territorial existentes (por ejemplo; expresa, troncal, colectora, servicio, local). Cada categoría de arco debe ser calibrada, lo cual hace necesario contar con un set de conteos para cada una de ellas.

La calibración de redes consiste en estimar los parámetros de la función de costo generalizado asociados a los arcos de una red en específico. En el caso del transporte privado, las funciones de costo tienen la siguiente forma <sup>2</sup>

$$t(f) = t_0 * \left[ 1 + \alpha \left( \frac{f}{K} \right)^\beta \right] \tag{1}$$

Donde  $t(f)$  es el tiempo de viaje en el arco para un nivel de flujo  $f$  en el arco,  $t_0$  es el tiempo de viaje a flujo libre,  $K$  es la capacidad del arco, mientras que  $\alpha$  y  $\beta$  son parámetros que deben ser estimados para cada categoría de arco que haya sido definida (un par de parámetros para cada categoría).

La siguiente figura muestra la forma esperada que debiesen tener estas curvas y la relación que debiese existir entre las distintas categorías de arcos.



**Figura 1** Formas esperadas de las curvas flujo-demora

Fuente: MESPE (SECTRA, 2008)

Por otro lado, la función objetivo a minimizar durante la calibración suele ser de la siguiente forma

$$\text{Min} \quad \sum_{a \in A_c} \{f_a^0 - \tilde{f}_a(x_1, x_2, \dots, x_n)\}^2 \tag{2}$$

Donde,  $A_c$  es el conjunto de arcos de la red vial sobre los que existen conteos de flujo vehicular,

<sup>2</sup> Este trabajo utiliza la forma funcional usada en los modelos ESTRAUS/VIVALDI. No se consideran análisis de formas funcionales alternativas en este trabajo. Sin embargo, la metodología desarrollada es aplicable a otros softwares de modelación u otras formas funcionales.

$f_a^0$  es el flujo vehicular observado en el arco  $a$  (conteo) y  $\tilde{f}_a$  es el flujo vehicular modelado en el arco (resultado del módulo de asignación de ESTRAUS/VIVALDI).

#### 4. DATOS

Para la realización del presente trabajo se solicitó a SECTRA acceso a las redes calibradas de forma reciente en tres ciudades del país. De esta forma, se tuvo acceso a las redes de modelación de las ciudades de San Antonio, Temuco y Valdivia, las cuales fueron calibradas en los estudios de SECTRA (2018), SECTRA (2014a) y SECTRA (2014b) respectivamente.

La Tabla 1 muestra las características principales de las redes de cada ciudad. Para este trabajo se consideró solamente el período punta mañana de cada una.

**Tabla 1** Características Redes de Modelación San Antonio, Temuco y Valdivia

	San Antonio	Temuco	Valdivia
N°Zonas	125	97	74
N°arcos	2,019	3,216	1,714
N°nodos	809	1,282	643
N°categorías de arco	5	6	6
Modelo	VIVALDI	ESTRAUS	VIVALDI

*Fuente: Elaboración Propia*

Por otro lado, la Tabla 2 y Figura 2 muestran las categorías de arco definidas para cada ciudad.

**Tabla 2** Categorías de Arco Redes de Modelación San Antonio, Temuco y Valdivia

Cat.	Tipo de Vía		
	San	Temuco	Valdivia
1	Autopista	Expresa	Troncal, no semaforizada, con tpte.público
2	Troncal	Troncal	Troncal y colector, semaforizada, con tpte.público
3	Colectora	Colectora	Colectora, sin tpte.público
4	Servicio	Servicio	Colectora, no semaforizada, con tpte.público
5	Local	Local	Servicio semaforizadas
6	-	Céntrica	Servicio no semaforizadas

*Fuente: SECTRA (2018), SECTRA (2014a) y SECTRA (2014b)*

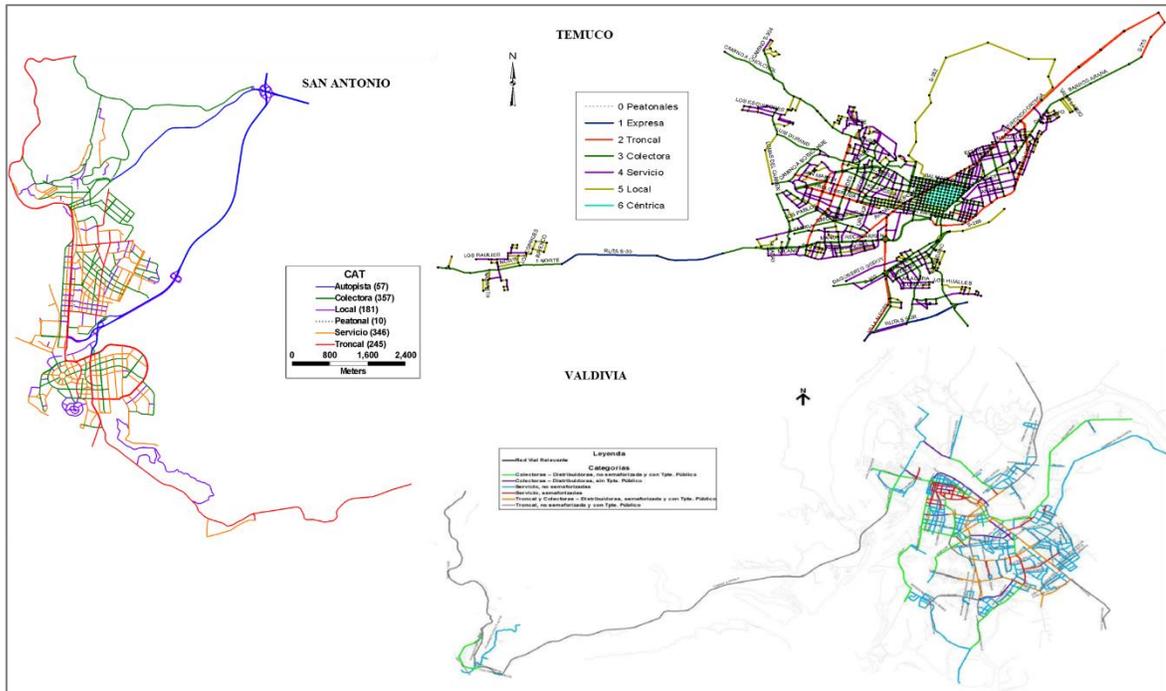
#### 5. MÉTODOLÓGÍA

El objetivo principal del presente trabajo apunta a mejorar la calidad de la calibración del módulo de asignación de transporte privado de los modelos ESTRAUS/VIVALDI. Para lo anterior, se consideraron tres aspectos fundamentales: valor final de la función objetivo (FO), consistencia de los parámetros según lo esperado por tipo de vía (Figura 1), y tiempo de calibración.

Para realizar el análisis comparativo, en este trabajo se seleccionaron cuatro algoritmos de optimización: Hooke & Jeeves, y tres algoritmos alternativos de búsqueda global: Nelder-Mead, Simulated Annealing y Particle Swarm. Estos cuatro algoritmos fueron implementados en Julia Language (versión 1.1.0) y se definieron tres etapas de análisis:

- Etapa 1: Comparación de Escenarios
- Etapa 2: Análisis de Consistencia de Algoritmos
- Etapa 3: Comparación Final

Las dos primeras etapas fueron analizadas solamente con las redes de modelación de la ciudad de San Antonio mientras que la tercera etapa consideró el análisis de las tres ciudades disponibles: San Antonio, Valdivia y Temuco.



**Figura 2** Categorías de Arco Redes de Modelación San Antonio, Temuco y Valdivia  
 Fuente: SECTRA (2018), SECTRA (2014a) y SECTRA (2014b)

### 5.1 Etapa 1: Comparación de Escenarios

En la primera etapa se buscó estudiar la relevancia en cada algoritmo del punto de partida que sea usado en una calibración. Esto, debido a que los resultados del algoritmo de Hooke & Jeeves, al ser un algoritmo de búsqueda local, dependen fuertemente del vector inicial desde donde comienza la calibración. Se definieron los siguientes tres escenarios:

- **Escenario Teórico:** corresponde a iniciar la calibración con un vector acorde a lo esperado según cada tipo de vía.
- **Escenario Calibrado:** corresponde a iniciar la calibración con un vector obtenido de la calibración del escenario teórico. Esto con el fin de comprobar el comportamiento de los algoritmos frente a un posible mínimo local o global, para este escenario se utiliza el vector encontrado por el algoritmo Nelder-Mead en el escenario teórico.

- **Escenario Extremo:** corresponde a un vector inicial con parámetros fuera de los rangos esperados. Esto con el fin de corroborar la robustez de cada uno de los algoritmos, es por esto por lo que el vector de parámetros inicial en este escenario contiene como valor el extremo superior de lo recomendado, es decir, el valor 9 en cada uno de los parámetros.

## 5.2 Etapa 2: Análisis de Consistencia de Algoritmos

Los conteos recogidos en terreno no son necesariamente una solución factible del módulo de asignación de ESTRAUS/VIVALDI. Lo anterior debido a inconsistencias o errores propios de una medición en terreno y a la simplificación propia de una red de modelación.

En esta etapa realizó un proceso inverso, el cual, partiendo de una solución de flujos asignados entregada por el modelo, se trata de recuperar los parámetros conocidos usados para obtener dicha solución. El objetivo principal de esta etapa es verificar cuál de los algoritmos tiene una mayor consistencia y capacidad para recuperar una solución conocida y que se sabe factible.

Para ello primero se definió la solución conocida  $(\alpha^*, \beta^*)$  y se ejecutó el módulo de asignación del modelo. Los flujos obtenidos del proceso de asignación se consideran como las observaciones o conteos disponibles para el proceso de calibración. Posteriormente se procedió a realizar la calibración desde algún punto inicial alternativo  $(\alpha', \beta')$  para poder recuperar el vector original  $(\alpha^*, \beta^*)$ .

Teniendo los resultados para cada uno de los algoritmos se evaluó cual algoritmo logró una mayor cantidad de aciertos, donde un acierto es considerado como la obtención del parámetro más cercano al parámetro original. Por definición, solo un algoritmo logra el acierto de cada parámetro.

## 5.3 Etapa 3: Comparación Final

El objetivo de esta etapa es comparar de manera final la calidad de la calibración entre el algoritmo de Hooke & Jeeves y los algoritmos alternativos propuestos en el presente trabajo. Para seleccionar qué algoritmos y de qué forma debiesen ser usados se definieron tres estrategias a partir de los resultados obtenidos en las dos etapas previas, dos de las cuales usaron una combinación de algoritmos.

# 6. RESULTADOS

## 6.1 Resultados Etapa 1: Comparación de Escenarios

En esta etapa se realizó la calibración de los parámetros del transporte privado de la ciudad de San Antonio utilizando los cuatro algoritmos implementados y los tres escenarios definidos: teórico, calibrado y extremo.

Cada uno de estos algoritmos posee especificaciones propias. La siguiente tabla muestra las especificaciones definidas para cada algoritmo en esta etapa.

**Tabla 3** Especificaciones de Algoritmos para Etapa I

	Hooke & Jeeves	Nelder-Mead	Simulated Annealing	Particle Swarm
Max nº de iteraciones*	200	1,000	47,000	1,000
Cota Superior	—	—	9.0	9.0
Cota Inferior	—	—	0.0	0.0
Magnitud de Exploración ( $\delta$ )	1.0	—	—	—
Disminución de Magnitud ( $\Delta$ )	0.5	—	—	—
Criterio de Salida	0.005	—	—	—

\*El número de iteraciones fue definido según pruebas iniciales donde se determinaron valores que permitieran obtener resultados razonables

Fuente: Elaboración Propia

La siguiente tabla muestra los resultados obtenidos para cada escenario analizado. Como se puede observar, el mayor porcentaje de mejora se logra mediante el algoritmo Simulated Annealing, pero con un costo de tiempo bastante alto en comparación con el resto de los algoritmos. Por otro lado, Particle Swarm y Nelder-Mead logran un porcentaje de mejora bastante competitivo con Simulated Annealing con un tiempo de calibración menor a la mitad que dicho algoritmo. En el caso del escenario calibrado ni Hooke & Jeeves ni Nelder-Mead logran mejorar significativamente la calibración, sin embargo, en el caso de Nelder-Mead se debe tener presente que la situación base de este escenario corresponde a la entregada por dicho algoritmo, por lo que es razonable que no se logren mejoras. Es interesante notar que, en el escenario extremo es donde se obtiene la mayor diferencia entre los algoritmos alternativos propuestos y Hooke & Jeeves.

**Tabla 4** Resultados Etapa I

Algoritmo	Escenario Teórico			Escenario Calibrado			Escenario Extremo		
	Tiempo	F.O	% mejora	Tiempo	F.O	% mejora	Tiempo	F.O	%
Base	-	2,466,000	-	-	2,190,000	-	-	3,124,000	-
Hooke & Jeeves	1.5	2,244,162	9%	0.9	2,184,000	0.3%	0.8	3,071,000	1.7%
Nelder-Mead	1.3	2,190,000	11%	1.1	2,180,000	0.4%	1	2,309,000	26%
Particle Swarm	10	2,095,000	15%	11	2,042,000	6.7%	10	2,084,000	33%
Simulated	24	1,999,000	19%	23	1,999,000	8.7%	24	2,084,000	33%

Fuente: Elaboración Propia

Al observar el comportamiento en los distintos escenarios se comprueba la tendencia que tiene el algoritmo de Hooke & Jeeves a quedar atrapado en mínimos locales, entregando valores de función objetivo muy distintos según el punto de partida utilizado. En el caso de los algoritmos alternativos revisados se observa una menor dependencia del punto de partida escogido. Por otro lado, los resultados de esta etapa sugieren que el algoritmo Particle Swarm logra los mejores resultados considerando un tiempo no excesivo de calibración.

## 6.2 Resultados Etapa 2: Análisis de Consistencia de Algoritmos

En primera instancia se probaron los algoritmos implementados con las mismas características de la etapa anterior. La siguiente tabla muestra los resultados obtenidos donde se puede ver que el algoritmo Nelder-Mead logra la mayor cantidad de aciertos.

**Tabla 5** Resultados Etapa II primer análisis

Algoritmo	Tiempo	F.O	Aciertos*
Hooke & Jeeves	2		0
Nelder-Mead	1		5
Particle Swarm	22		3
Simulated Annealing	15		2

*\*Nota: El total de parámetros a estimar es 10 pues la ciudad de San Antonio posee cinco categorías de arco*

*Fuente: Elaboración Propia*

Considerando los buenos resultados que se obtuvieron en la Etapa I de los algoritmos Particle Swarm y Simulated Annealing, se realizó un segundo análisis a partir de la siguiente hipótesis: “Debido a la amplia capacidad de búsqueda que tienen estos dos algoritmos, si se acota el campo de búsqueda de estos, van a poder mejorar la solución entregada”.

Por lo anterior, se determinó que el rango entre la cota inferior y superior para cada parámetro sería de 6 unidades de diferencia respecto al valor del parámetro (conocido para esta etapa) y para poder realizar una comparación más coherente frente a Hooke & Jeeves y Nelder-Mead, se restringió el tiempo de búsqueda de ambos algoritmos disminuyendo el número de iteraciones (1500 en Simulated Annealing y 200 en Particle Swarm). La siguiente tabla muestra como variaron los resultados al introducir las modificaciones mencionadas en las especificaciones de los algoritmos.

**Tabla 6** Resultados Etapa II con restricción de cotas

Algoritmo	Tiempo	F.O	Acierto
Hooke & Jeeves	2	15,643	0
Nelder-Mead	1	1,528	0
Simulated Annealing	22	2,216	0
Particle Swarm	15	5,641	0
Simulated Annealing	2	1,497	2
Particle Swarm (Cotas)	1.2	259	8

*Fuente: Elaboración Propia*

Como se puede observar, los resultados obtenidos con restricción de cotas en los algoritmos Simulated Annealing y Particle Swarm modifican totalmente la situación obtenida cuando no se incorpora la restricción. Este resultado confirma que, dado que ambos algoritmos poseen una mayor capacidad de búsqueda, requieren de más tiempo para poder encontrar un mínimo global, sin embargo, si se les entrega rango acotado de búsqueda, el tiempo de calibración se reduce significativamente y se vuelve competitivo con los tiempos de los otros dos algoritmos. Estos resultados sugieren la conveniencia de usar una combinación de algoritmos durante la calibración.

### 6.3 Resultados Etapa 3: Comparación Final

En esta última etapa de análisis se realiza la comparación final entre los algoritmos alternativos propuestos y Hooke & Jeeves. Para esto se analizan las tres ciudades consideradas en el análisis.

De los resultados obtenidos en las dos etapas previas de análisis, fue posible diseñar tres estrategias a ser estudiadas en la etapa final:

- **Estrategia I** (combinación de algoritmos): Calibrar en primera instancia con el algoritmo Nelder-Mead, para posteriormente, a partir de dicho resultado, utilizar Simulated Annealing con restricción de cotas. La restricción de cotas consiste en tener una amplitud de búsqueda de 6 unidades por parámetro desde la solución encontrada por Nelder-Mead.
- **Estrategia II** (combinación de algoritmos): Calibrar en primera instancia con el algoritmo Nelder-Mead, para posteriormente, a partir de dicho resultado, utilizar Particle Swarm con restricción de cotas. La restricción de cotas consiste en tener una amplitud de búsqueda de 6 unidades por parámetro desde la solución encontrada por Nelder-Mead.
- **Estrategia III:** Utilizar el algoritmo Particle Swarm sin restricción de cotas.

Las especificaciones de cada algoritmo se mantienen según las descritas en las etapas previas, considerando restricción de cotas para Simulated Annealing y Particle Swarm. En el caso de Particle Swarm, se extendió el número de iteraciones a 500 con el fin de aumentar la factibilidad de encontrar una mejor solución y mejorar el desempeño del algoritmo. Por otro lado, en el caso del algoritmo de Hooke & Jeeves, al ser un algoritmo de búsqueda local, se consideró la realización de al menos 3 calibraciones desde distintos puntos de partida, con el fin de seleccionar la mejor solución obtenida, mientras que, en el caso de las tres estrategias definidas, al basarse en algoritmos de búsqueda global, solo se realizó una calibración a partir de un punto inicial.

La comparación de esta etapa se realizó por medio del indicador GEH, el coeficiente de correlación de Pearson ( $R^2$ ) entre flujos observados y modelados, el valor de la función objetivo, y el tiempo de calibración. El indicador GEH es generalmente usado en ingeniería de tráfico y modelación de transporte para comparar dos conjuntos de volúmenes de tránsito vehicular. El criterio de aceptación suele ser un GEH promedio de la red modelada menor a 5 y no permitir valores mayores a 10 en arcos importantes de la red (MDS-SECTRA, 2013).

$$GEH = \sqrt{\frac{2*(O-C)^2}{O+C}} ; O: \text{flujos observados}; C: \text{flujos asignados} \quad (3)$$

En la Tabla 7 se muestran los resultados obtenidos. Por otro lado, con respecto a los parámetros obtenidos, ningún algoritmo logró una consistencia total para todas las categorías de arco calibradas. Sin embargo, esta consistencia nunca ha estado garantizada a priori, por lo que generalmente el proceso de calibración requiere un modelador experto que pueda evaluar el ajuste de las curvas obtenidas de forma de contar con un modelo que reproduzca de buena forma los flujos observados sin contradecir el comportamiento esperado de cada categoría de arco. Esto último es de suma relevancia dado que el objetivo último del modelo construido no es meramente reproducir la situación observada sino el uso predictivo a futuro.

La Tabla 8 muestra los parámetros obtenidos para el caso de la ciudad de Temuco para las categorías expresa, troncal, colectora y servicio, que suelen ser las calles más relevantes de la red

de modelación y dónde es más fácil identificar el comportamiento esperado<sup>3</sup>. Se puede observar que Hooke & Jeeves y las Estrategias II y III muestran parámetros razonables si se excluye la categoría troncal, la cual no muestra resultados satisfactorios en ninguno de los casos (con respecto a los rangos esperados de los parámetros según la Figura 1). Tal como fue discutido previamente es de esperar que categorías de vías “menores” posean alfas mayores y betas pequeños, mientras que categorías troncales o utopistas presenten alfas pequeñas y betas mayores.

**Tabla 7** Resultados Etapa III

Ciudad	Algoritmo	GEH	R <sup>2</sup>	Tiempo	FO	% mejora
San Antonio	Hooke & Jeeves	6.00	0.80	1.5x4=6	2,224,000	-
	Estrategia 1	5.74	0.81	20	2,000,000	10.1%
	Estrategia 2	5.79	0.81	14	2,041,656	8.2%
	Estrategia 3	5.91	0.80	9	2,094,526	5.8%
Valdivia	Hooke & Jeeves	6.69	0.80	4 x 4 =16	2,483,259	-
	Estrategia 1	6.42	0.82	15	2,232,298	10.1%
	Estrategia 2	6.33	0.83	28	2,202,112	11.3%
	Estrategia 3	6.36	0.83	24	2,168,607	12.7%
Temuco	Hooke & Jeeves	6.38	0.84	2.5 x 4	7,738,297	-
	Estrategia 1	6.36	0.83	7	7,600,000	1.8%
	Estrategia 2	6.29	0.85	13	7,582,421	2.0%
	Estrategia 3	6.22	0.83	12	7,570,720	2.2%

Fuente: Elaboración Propia

**Tabla 8** Parámetros Etapa III ciudad de Temuco

Curva	Tipo	Hooke &		Estrategia		Estrategia		Estrategia	
		alfa	beta	alfa	beta	alfa	beta	alfa	beta
1	Expresa	2.20	6.00	7.06	3.60	2.12	2.72	3.66	2.20
2	Troncal	8.93	2.45	5.94	1.71	6.01	1.66	8.23	1.98
3	Colectora	6.75	2.72	3.52	1.54	3.67	1.50	3.29	1.36
4	Servicio	7.84	2.85	4.59	1.78	5.36	1.78	5.69	1.97

Fuente: Elaboración Propia

## 7. CONCLUSIONES

Los resultados obtenidos en la primera etapa de análisis comprueban la tendencia del algoritmo de Hooke & Jeeves a caer en mínimos locales, entregando valores de función objetivo muy distintos según el punto de partida utilizado. En el caso de los algoritmos alternativos revisados se observa una menor dependencia del punto de partida escogido. Por otro lado, los resultados de esta etapa sugieren que el algoritmo Particle Swarm logra los mejores resultados considerando un tiempo de calibración razonable.

De la segunda etapa de análisis se concluye que, si bien la potencia para recuperar una solución conocida por parte de los algoritmos Particle Swarm y Simulated Annealing es mayor que Hooke

<sup>3</sup> Si bien la categoría céntrica también es relevante, dada su naturaleza, es más difícil establecer el rango de los parámetros esperados con respecto a las categorías restantes.

---

& Jeeves y Nelder-Mead, la calidad de la calibración depende en gran parte de acotar el rango de búsqueda. De esta forma, si los algoritmos conocen dónde buscar, son capaces de encontrar una buena solución en un tiempo competitivo con respecto a Hooke & Jeeves. Por lo anterior, de esta etapa se desprenden dos estrategias para analizar en la etapa final, ambas basadas en una búsqueda inicial con el algoritmo Nelder-Mead para posteriormente realizar una segunda búsqueda de forma acotada, ya sea con Simulated Annealing o con Particle Swarm.

En el caso de la Etapa III de análisis se concluye que las tres estrategias logran una calibración con mejores indicadores que el algoritmo de Hooke & Jeeves. Si bien ningún algoritmo logró una consistencia total para todas las categorías de arco calibradas, esta consistencia ha sido siempre difícil de lograr de forma absoluta y suele requerir la intervención de un modelador experto con conocimiento de la ciudad (quien puede, por ejemplo, agrupar categorías, heredar parámetros, ajustar parámetros, eliminar conteos sospechosos, etc.). Finalmente, una forma de apoyar esta tarea sería ajustar la implementación de los algoritmos de manera de forzar la consistencia esperada, por ejemplo, volviendo a calibrar la categoría cuestionada con un rango de búsqueda menor.

### AGRADECIMIENTOS

Agradecemos a SECTRA por poner a disposición los datos que permitieron la realización del presente trabajo, en particular al Sr. Alan Thomas, quien en su momento ejercía el cargo de Coordinador del Área Metodológica de dicha institución.

### REFERENCIAS

- Gao, F., y Han, L. (2012) Implementing the Nelder-Mead Simplex algorithm with adaptive parameters. *Computational Optimization and Applications* **51**, 259-277.
- Goffe, W.L., Ferrier, G.D. y Rogers, J. (1994) Global optimization of statistical functions with simulated annealing. *Journal of Econometrics* **60**, 65-99.
- Goffe, W.L. (1996) SIMANN: a global optimization algorithm using simulated annealing. *Studies in Nonlinear Dynamics and Econometrics* **1**, 1-9.
- Hooke R. y Jeeves T.A. (1961) "Direct search" solution of numerical and statical problems. *Journal of the Association of Computer Machinery* **8**, 212-229.
- Kennedy, J. y Eberhart, R (1995) Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks* **4**, 1942-1948.
- MDS-SECTRA (2013) Manual de Evaluación Social de Proyectos de Vialidad Urbana: MESPIVU. Ministerio de Desarrollo Social, Santiago
- Nelder, J.A. y Mead, R. (1965) A Simplex method for function minimization. *Computer Journal* **7**, 308-313.
- SECTRA (2008) Actualización Metodología Análisis de Sistema de Transporte en Ciudades Medianas y Grandes (MESPE). Programa de Vialidad y Transporte Urbano, Santiago <http://www.sectra.gob.cl/metodologias/mespe.htm>.

SECTRA (2014a) Actualización Plan de Transporte Temuco y Desarrollo de Anteproyecto, I Etapa. Programa de Vialidad y Transporte Urbano, Santiago <http://www.sectra.gob.cl/biblioteca/detalle1.asp?mfn=3434>

SECTRA (2014b) Actualización Plan de Transporte Valdivia y desarrollo de anteproyecto, I Etapa. Programa de Vialidad y Transporte Urbano, Santiago <http://www.sectra.gob.cl/biblioteca/detalle1.asp?mfn=3290>

SECTRA (2018) Actualización Diagnóstico del S.T.U de San Antonio. Programa de Vialidad y Transporte Urbano, Santiago <http://www.sectra.gob.cl/biblioteca/detalle1.asp?mfn=3478>

Zhan, Z. H., Zhang, J., Li, Y. y Chung, H.S.H. (2009) Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **39**, 1362-1381.