

EVALUACIÓN DE TÉCNICAS PARA LA CALIBRACIÓN DEL
MODELO DE MICROSIMULACIÓN: TRANSMODELER

Presentado por:

Juan Sebastián Arias Hernández

Presentado como requisito:

Para optar por el título de Magister en Investigación Operativa y Estadística.

Director:

Dr. Mauricio Granada Echeverri.

Jurados:

MSc David Alejandro Jiménez

MSc Orlando Antonio Sabogal

Maestría en Investigación Operativa y Estadística

Facultad de Ingeniería Industrial

Universidad Tecnológica de Pereira

Pereira, septiembre de 2018.

Dedicatoria:

A Adriana, por hacerme feliz con todo su amor. Por mostrarme que hay una vida más allá de las ecuaciones y de la programación. Por ser el código fuente de mi vida.

Agradecimientos:

A mi esposa, por todo su esfuerzo, sacrificio, ideas y paciencia puestas en la realización del presente trabajo.

A mis padres, hermanos y sobrinos por ser fuente de alegría en mi vida.

A mi director por su ayuda y colaboración.

A mis compañeros y amigos, David y Orlando, por compartir propuestas e ideas para hacer de este un mejor trabajo; por las charlas de fútbol, política y ciencia.

A Fabian y Andrés, y en general al equipo de simulación por todo el apoyo y disposición para solventar las dudas y problemas técnicos que surgieron durante la investigación.

Al grupo de investigación Sirius, por todos los momentos y experiencias vividas.

A la Facultad de Ingeniería Industrial por el apoyo brindado para la realización de la maestría.

A la Maestría en Investigación Operativa y Estadística por el esfuerzo puesto para hacer de este un programa de alta calidad.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	6
ÍNDICE DE TABLAS	7
GLOSARIO	8
RESUMEN	10
ABSTRACT	14
1. Aspectos Preliminares	17
1.1. Introducción	17
1.2. Planteamiento del Problema.	19
1.3. Pregunta de Investigación	20
1.4. Objetivo General	20
1.4.1. Objetivos Específicos	20
1.5. Justificación	21
2. El problema de calibración.....	23
2.1. Revisión del estado del arte.	23
3. Marco Teórico.	31
3.1. Optimización basada en simulación o Simulation-Optimization.	32
3.2. Proceso de calibración.	34
3.3. Algoritmo de optimización.	37
3.3.1. Partícula Swarm.....	37
3.3.2. SPSA: Simultaneous Perturbation Stochastic Approximation	41
3.3.3. Algoritmo genético.....	44
3.4. Marco teórico para la realización de las pruebas estadísticas.....	51
3.4.1. Teorema del límite central.....	51
3.4.2. Estimadores.	52
3.4.3. Intervalos de Confianza.....	53
3.5. Tamaño de la muestra.	53
4. Metodología propuesta.....	54
4.1. Delimitación y descripción de la zona de estudio.....	54
4.2. Recolección y tratamiento de la información utilizada.	56
4.3. Modelo de microsimulación.	58

4.3.1.	Modelamiento de la red	58
4.3.2.	Sentidos y giros permitidos.	58
4.3.3.	Semáforos y sensores.	59
4.3.4.	Matriz Origen – Destino, OD.	60
4.4.	Software, modelos y parámetros de calibración.	62
4.4.1.	Herramienta computacional: Gauss.....	62
4.4.2.	Modelo: Car Following	65
4.4.3.	Distancias de las Paradas.....	66
4.4.4.	Headway.....	67
4.5.	Variables de campo para calcular una medida de discrepancia (MoP Measures of performance).....	68
4.6.	Función Objetivo.	69
4.7.	Descripción detallada de los algoritmos implementados.....	70
4.7.1.	Algoritmo genético.	70
4.7.2.	Enjambre de partículas.	74
4.7.3.	SPSA.....	76
4.8.	Características para tener en cuenta al realizar calibración.	78
4.8.1.	Semilla.....	78
4.8.2.	Cantidad de réplicas.....	78
4.8.3.	El periodo de calentamiento.	79
5.	Resultados	79
5.1.	Resultados generales.....	80
5.2.	Búsqueda realizada por los algoritmos	83
5.2.1.	Resultados para el SPSA	85
5.2.2.	Resultados para el Algoritmo genético.....	86
5.2.3.	Resultados para el PSO	87
5.2.4.	Búsqueda realizada por los algoritmos.....	89
5.3.	Gráficos de convergencia.....	91
5.4.	Evaluación de los resultados.....	93
5.4.1.	Valores numéricos de los resultados obtenidos.....	93
5.4.2.	Gráficos de densidad.	95
5.4.3.	Gráficos con los intervalos de confianza.....	97
5.4.4.	Indicador de convergencia.....	98
5.5.	Comparación de los resultados con los reportados en la literatura.....	99
6.	Conclusiones y trabajos futuros	101
6.1.	Conclusiones.....	101

6.2. Trabajos Futuros	103
Bibliografía.....	104
Anexos.....	110

ÍNDICE DE FIGURAS

Figura 1. Árbol resultado del estado del arte en ToS. Fuente: (Robledo, 2014).....	23
Figura 2. Proceso de simulación. Fuente: (Carson, 1997).....	32
Figura 3. Optimización basada en simulación. Fuente: Elaboración Propia.....	33
Figura 4. Proceso de Calibración. Fuente: Elaboración Propia.....	36
Figura 5. Estructura del PSO. Fuente: Elaboración Propia.....	41
Figura 6. Estructura del SPSA. Fuente: Elaboración Propia.....	43
Figura 7. Función Griewank. Adaptada de: (Eiben, 2008).....	45
Figura 8. Pasos para implementar AG. Fuente: Elaboración propia.....	47
Figura 9. Estructura del AG Fuente: Elaboración Propia.....	51
Figura 10. Zona bajo estudio. Fuente: open street map.....	55
Figura 11. Posición de los sensores. Fuente: Grupo de investigación Sirius.....	59
Figura 12. Orígenes – Destinos. Fuente: Grupo de investigación Sirius.....	60
Figura 13. Modelo de microsimulación implementado. Fuente: Grupo de investigación Sirius.....	61
Figura 14. Arquitectura de Gauss. Fuente: (Jiménez, 2016).....	63
Figura 15. Pasos para implementar un método en Gauss. Fuente: Elaboración propia.....	65
Figura 16. Car Following. Adaptado de: (Caliper, 2011).....	65
Figura 17. Headway. Adaptado de: (Caliper, 2011).....	67
Figura 18. Límites de tiempo Headway. Adaptado de: (Caliper, 2011).....	67
Figura 19. Implementación del AG. Fuente: Elaboración Propia.....	71
Figura 20. Implementación PSO. Fuente: Elaboración Propia.....	75
Figura 21. Implementación SPSA. Fuente: Elaboración Propia.....	78
Figura 22. Resultados generales de los algoritmos implementados. Fuente: Elaboración Propia.....	80
Figura 23. Perspectiva parámetro beta. Fuente: Elaboración Propia.....	82
Figura 24. Perspectiva parámetro gamma. Fuente: Elaboración Propia.....	82
Figura 25. Muestra de los resultados. Fuente: Elaboración Propia.....	83
Figura 26. Valores encontrados por los algoritmos. Fuente: Elaboración Propia.....	84
Figura 27. Coordenadas paralelas SPSA. Fuente: Elaboración Propia.....	86
Figura 28. Coordenadas paralelas AG. Fuente: Elaboración Propia.....	87
Figura 29. Coordenadas paralelas PSO. Fuente: Elaboración Propia.....	88
Figura 30. Mapa de calor de los algoritmos implementados. Fuente: Elaboración Propia.....	90
Figura 31. Convergencia de los Algoritmos. Fuente: Elaboración Propia.....	92
Figura 32. Convergencia de los Algoritmos con Filtro. Fuente: Elaboración Propia.....	93
Figura 33. Gráficos de densidad. Fuente: Elaboración Propia.....	95
Figura 34. Box plot de los resultados. Fuente: Elaboración Propia.....	96
Figura 35. Intervalos para la diferencia de medias. Fuente: Elaboración Propia.....	98
Figura 36. Diferencia de Medias para el Área. Fuente: Elaboración Propia.....	99

ÍNDICE DE TABLAS

Tabla 1. Evolución vs. Optimización.	45
Tabla 2. Resumen de la información utilizada.	56
Tabla 3. Intersecciones semaforizadas.	59
Tabla 4. Orígenes – Destinos.....	61
Tabla 5. Parámetros por calibrar.....	68
Tabla 6. Configuración implementada en AG.....	74
Tabla 7. Valores configuración PSO.	76
Tabla 8. Valores de la configuración del SPSA.	78
Tabla 9. Valores de los resultados.	94
Tabla 10. Indicador de convergencia.....	98
Tabla 11. Resumen de trabajos futuros propuestos	104

GLOSARIO

Accesibilidad: habilidad de los individuos de alcanzar oportunidades (bienes, servicios, actividades, etc). La facilidad que las personas tienen para acceder a estos bienes y servicios.

Algoritmo: conjunto de procedimientos lógicos que garantizan encontrar una solución a un problema, en un tiempo que se considera aceptable.

Calibración: proceso de encontrar los mejores valores de los parámetros para los submodelos de un microsimulador de tráfico.

Capacidad de la red: cantidad máxima de vehículos que puede soportar una red, bajo unas condiciones establecidas.

Demanda: en transporte, se refiere a la cantidad de vehículos que desean acceder a la red en lugar(es) particulares.

Flujos: en transporte es la tasa a la que pasan los vehículos en un punto determinado.

Heurísticas: son estrategias que permiten encontrar buenas soluciones, pero no garantizan que sea la óptima. Estos procedimientos se emplean para problemas específicos.

Interfaz: conexión funcional y lógica entre dos programas que permite transmitir información en doble vía en tiempo real para resolver un problema.

Metaheurísticas: son estrategias generales que pueden ser aplicadas a diferentes tipos de problemas. Al igual que las heurísticas no garantizan una respuesta óptima.

Microsimulador de tráfico: es una herramienta computacional que permite simular de manera detallada el comportamiento individual de los vehículos y sus interacciones con el entorno. Caracterizando por medio de submodelos diferentes comportamientos como la aceleración, la velocidad, cambios de carril, elección de ruta etc.

Movilidad: habilidad que se tiene para desplazar de un lugar a otro personas, bienes y servicios en la mayor brevedad posible.

Oferta: en transporte, hace referencia a toda la infraestructura construida para la movilidad vehicular.

Parámetros: argumentos de la función que afectan el comportamiento de esta pero no se asumen como variables independientes ya que no es de interés en el modelamiento.

Red vial: infraestructura física que permite el movimiento vehicular de un lugar a otro de personas, bienes o servicio.

Simulación Computacional: técnica que permite abstraer la realidad mediante un ordenador con el fin de imitar su operación y comprender como funciona el sistema. Adicionalmente como afecta el comportamiento de este al modificar información del modelo.

RESUMEN

Para hacer frente a la sostenibilidad en el transporte y la movilidad, son necesarias herramientas que evidencien una adecuada toma de decisiones y permitan comprender por medio de la experimentación las dinámicas urbanas propias de cada región; de tal forma que sea posible evaluar el impacto de la implementación de políticas de infraestructura, tecnología, entre otras. Actualmente, uno de los enfoques reconocidos para realizar esta tarea en la administración de las operaciones de tráfico y movilidad son las tecnologías ITS (acrónimo en inglés de Intelligent Transport System), donde la microsimulación se destaca como una herramienta importante dado que evalúa el impacto de diferentes medidas y estrategias de movilidad antes de ser implementadas en la realidad, proceso valioso ya que ahorra tiempo, dinero e incomodidades (Rodrigues, 2014).

Para que los resultados obtenidos del microsimulador sean confiables, se debe garantizar que el modelo represente las condiciones urbanas y los patrones de conducción particulares de la zona bajo estudio; puesto que como se expone ampliamente en la literatura dichos patrones pueden variar, incluso entre ciudades cercanas (Treiber, 2013). El proceso de hallar los mejores valores de los parámetros para que los microsimuladores sean ajustados a las condiciones locales es conocido como calibración.

En la presente investigación se aborda el problema de calibrar el modelo de microsimulación Transmodeler para la zona de estudio comprendida entre carreras 13 y 14 con calles 18 y 25 en la ciudad de Pereira, Risaralda, Colombia. Zona caracterizada por el gran flujo vehicular y a que contará con la primera línea de cable aéreo de la ciudad (Alcaldía Pereira, 2017) por lo que debe de estar modelada adecuadamente para evaluar

futuras políticas como por ejemplo la implementación de semáforos inteligentes que lleven a un mejor uso de la red.

Para lograr lo anterior, se implementan tres estrategias en la solución del problema de calibración. La primera de ellas es conocida como una aproximación estocástica mediante una perturbación simultánea SPSA (siglas en inglés de *Simultaneous Perturbation Stochastic Approximation*) la cual es ampliamente abordada en la literatura; esta realiza una aproximación del gradiente sin requerir de una evaluación explícita de la derivada, algo que representa una enorme ventaja en el proceso de calibración. La segunda es un algoritmo genético, en adelante AG, y aunque esta aparece con frecuencia en la literatura el presente trabajo realiza una codificación real, siendo esta una codificación poco utilizada en el área de estudio. Esta metaheurística se basa en la evolución y por ende busca mejorar una población al realizar operaciones de cruzamiento y mutación sobre los individuos, estos representan a su vez diferentes soluciones del problema, de manera que la solución mejora con las generaciones. El tercer algoritmo desarrollado fue un enjambre de partículas o PSO (siglas en inglés para *Particles Swarm Optimization*) el cual está inspirado en el comportamiento social de ciertas especies como aves, peces y abejas. Este ha sido poco utilizado para calibrar modelos de microsimulación y por tal motivo desarrollado en la presente investigación.

La comparación de las estrategias se realiza con datos reales tomados en la zona de estudio, algo que vale la pena resaltar de la investigación ya que generalmente los investigadores no cuentan con información de campo por lo que recurren a datos sintéticos para realizar el proceso de calibración. Esta información consiste de la matriz origen – destino de la zona, así como flujos en ciertos puntos de interés que permiten comparar las salidas del simulador con lo observado en campo al modificar algunos parámetros del

modelo, por medio de estrategias inteligentes, de manera que permita minimizar la discrepancia entre las medidas reales y las simuladas; distancia conocida como *goodness of fit* en la literatura, esto se realiza con el objetivo de que el modelo de microsimulación se ajuste a la realidad bajo estudio.

Para el caso de estudio se calibran los parámetros beta y gamma del modelo *car following*, así mismo se ajusta el parámetro que modela la distancia que se deja con respecto al vehículo de adelante perteneciente al modelo *distance gap* y por último se trabaja el tiempo que se deja al auto de adelante cuando los vehículos se encuentran en movimiento del modelo *time headway*. Los parámetros abordados fueron seleccionados considerando que afectan de manera directa la capacidad de la red (Dowling, 2004) y a los resultados obtenidos en (Jiménez, 2016).

Los hallazgos de la investigación muestran que los valores de los parámetros encontrados a partir de los tres métodos implementados presentan mejores comportamientos que los valores que por defecto trae Transmodeler. Particularmente, el algoritmo genético encuentra mejores resultados para el caso de estudio. El enjambre de partículas explora más el espacio de soluciones, aunque en algunos casos mostró que no logra escapar de regiones con baja calidad. Frente al SPSA se concluye que requiere tratamientos adicionales para mejorar sus resultados, los cuales son planteados en trabajos futuros. Para las implementaciones y tratamientos antes descritos se utilizaron los lenguajes C#, R y *postgres sql*.

El presente estudio se enmarca en la línea de investigación en ITS que tiene el grupo Sirius de la Universidad Tecnológica de Pereira. Este grupo cuenta con los software Vissim (PTV Group, 2018) y Transmodeler (Caliper, 2018) los cuales han sido utilizados para llevar a cabo diferentes estudios de movilidad en la ciudad de Pereira. La presente

investigación se articula con investigaciones previas realizadas por el grupo en el campo de calibración dando continuidad a un proceso que desarrolló un software para la calibración de Transmodeler (Jiménez, 2016); buscando aumentar los algoritmos implementados para realizar el proceso de optimización.

ABSTRACT

Nowadays many cities must face traffic congestion problems, this occurs because the use of the traffic network increase due mayor volume of traffic. This has an impact in many ways like: wasted time, air pollution, stress and others. One of the objective of traffic management is make an efficient use of network, to accomplish that, different decisions can take like: grow the ofert, use intelligent transport system (user information system, intelligent traffic light etc.) and so on.

To evaluate the decisions above microscopic traffic simulation model have significant attention because has the potential to perform computer experiments of a particular zone of study. This enable to traffic manager evaluate politics before carrying on to real world avoided waste resource like money and time.

To have reliable results, the microsimulator parameters must be adjusted. In consequence the model should represents the urban conditions and driving patterns of the area under study; since, these characteristics may vary, even among nearby cities (Treiber, 2013). The process of finding the best parameters for microsimulators is known as calibration.

The present study is part of the research line in ITS that has the Sirius group at the Universidad Tecnológica de Pereira. This group has microsimulation software such as Vissim (PTV Group, 2018) and Transmodeler (Caliper, 2018), which have been used to carry out different mobility studies in Pereira city. This thesis evaluates algorithms to calibrate some of the parameters in the Transmodeler microsimulation software for the area between 13th and 14th Avenue and streets 18th and 25th in the city of Pereira, Risaralda, Colombia.

To carry out the investigation, it was necessary to collect and analyze the information for the origin-destination matrix, as well as flows in certain points of interest. This information is used to compare the outputs of the simulator with those observed in the field.

This research implemented three strategies to the solution of the calibration problem. The first two are commonly referenced in the literature: genetic algorithms, AG and Simultaneous Perturbation Stochastic Approximation or SPSA. The AG is based on evolution and therefore seeks to improve an initial population by carrying out crossover and mutation operations on individuals, which in turn represent different solutions to the problem, so that the solution improves with the generations. The second one approximates the gradient, like finite differences, with the advantage of not requiring an analytical function.

The third algorithm implemented was a particles swarm and although in the state of the art it is little addressed for the problem under study is contemplated in this work, because its structure should allow to explore the space of solutions and converge faster than the genetic algorithms. The PSO as well as the AG are based on an initial population. However, the PSO looks for each particle to move in the space of solutions to the best position found by the swarm.

The parameters of the car following model, beta and gamma were calibrated, as well as the distance left with respect to the vehicle ahead which belong to the distance gap model and finally the time left to the car ahead where calibrated for headway model. The parameters addressed were selected considering that they affect straight the capacity of the network (Dowling, 2004).

The results obtained show that the values of the parameters found from the three implemented methods show better behavior than the default values from Transmodeler. Particularly, the genetic algorithm finds better results for the case study. However, the swarm of particles explores more the space of solutions increasing the possibility of escaping from local optimal. The conclusion for the SPSA is that this algorithm requires additional treatments to increase the quality of its results, therefore some are proposed in future works. For the implementations and treatments described above, the languages C #, R and postgres sql were used.

1. Aspectos Preliminares

1.1. Introducción

En la actualidad, grandes esfuerzos se llevan a cabo para mejorar la eficiencia de los sistemas de transporte debido a los impactos sociales, ambientales y económicos de éstos en el mundo. Estos esfuerzos abarcan áreas que van desde la planeación y accesibilidad hasta la evaluación de la infraestructura y de tecnologías que permitan un mejor uso de la red de transporte (Daamen, 2014a). En cuanto a esto último, el identificar cómo se puede comportar el sistema al implementar políticas como construir calles, puentes o hacer uso de Sistemas Inteligentes de Transporte ITS, es de gran utilidad práctica, ya que puede evaluar sí la política es adecuada o no en la realidad; evitando pérdidas de tiempo, dinero.

Realizar un modelamiento analítico del sistema de transporte es inmanejable debido a la complejidad de las relaciones entre los vehículos, los peatones, infraestructura vial, señalizaciones, sistemas inteligentes y de información a los usuarios; razón por la cual los simuladores de tráfico emergen como una alternativa de modelamiento. De acuerdo con (Daamen, 2014a) estos simuladores emulan el comportamiento que presentan los vehículos, por ejemplo, la manera como aceleran, la distancia que guardan con respecto al de adelante, la forma en que cambian de carril, si dan o no vía, etc. Cuando los simuladores caracterizan en detalle y particularmente el comportamiento de cada vehículo con respecto a su entorno reciben el nombre de microsimuladores.

Los microsimuladores generalmente contienen un conjunto de submodelos que controlan diferentes comportamientos, estos a su vez incluyen parámetros que han de ser ajustados con el fin de representar la realidad del lugar de investigación por medio del simulador. El proceso de ajustar los parámetros del modelo para mejorar la capacidad de

reproducir el comportamiento local de las condiciones de tráfico recibe el nombre de calibración; esta tarea debe desarrollarse antes de realizar cualquier estudio, de lo contrario los resultados obtenidos no son fiables.

La calibración de un modelo de microsimulación es una tarea compleja, por ello es preciso utilizar metaheurísticas que realicen una búsqueda inteligente de los mejores valores de los parámetros del microsimulador. Algunas características deseables en los algoritmos de búsqueda son: evitar óptimos locales y valles mientras explora de manera amplia el espacio de soluciones, y el procurar un número reducido de simulaciones ya que el tiempo y el esfuerzo computacional requeridos para realizar una simulación son altos y crecen con el tamaño de la red y los vehículos en ella.

En el presente estudio se comparan tres técnicas para la calibrar el modelo de microsimulación Transmodeler. Dos de ellas, AG y SPSA, son ampliamente abordadas en la literatura, aunque para la primera se realiza una codificación real que poco se utiliza en las investigaciones de calibración cuando se utiliza dicha técnica. Mientras la tercera, PSO, como se evidencia en el estado del arte, se ha utilizado pocas veces para calibrar un modelo de microsimulación. Se pretende entonces identificar cuál de ellas presenta más consistencia en búsqueda de la solución y cuál logra mejores resultados para la zona de estudio comprendida entre las carreras 13 y 14 con calles 18 y 25 en la ciudad de Pereira. Lugar de donde se obtuvo información de la matriz origen destino y flujos en ciertos puntos de interés.

Los principales aportes para resaltar de la presente investigación son: se utilizan datos reales para llevar a cabo el proceso, generalmente los investigadores no tienen acceso a esta información debido a la falta de recursos como el tiempo y dinero, por lo que son reemplazados por datos sintéticos. Por otro lado, se implementa un algoritmo poco utilizado

en la literatura: el PSO; de igual manera, se emplea un algoritmo genético con codificación real la cual ha sido poco abordada en calibración. Se realizan procedimientos para comparar resultados que en calibración se obvian como los intervalos de confianza; por último, se da continuidad un área que a nivel nacional ha sido poco estudiada y en la cual el grupo Sirius ha sido pionera en ciudades amables como Pereira.

Este documento tiene la siguiente estructura: En el resto de la sección 1 presenta y define el problema de investigación. En el numeral 2 se expone el estado del arte para el problema bajo estudio. En la sección 3 se introduce el marco teórico de las técnicas a implementadas. En el numeral 4 se muestra la metodología utilizada. En el 5 se exponen el análisis de resultados y la evaluación de las técnicas. Finalmente, en la sección 6 se presentan las conclusiones y trabajos futuros.

1.2. Planteamiento del Problema.

Los microsimuladores son software que permiten modelar la interacción existente entre los vehículos y la infraestructura de transporte (vías, señalizaciones, sistemas de información al usuario, semáforos inteligentes, etc.). Estos microsimuladores son utilizados para evaluar diferentes políticas que van desde intervenir físicamente la red, por ejemplo, construir una glorieta; hasta invertir el sentido de una calle o carrera con el fin de identificar si la solución propuesta mejora o no el sistema.

Ahora bien, para poder extrapolar las conclusiones que se obtengan de un microsimulador se debe garantizar que este represente la realidad de manera adecuada, tarea que no es trivial ya que estas herramientas están compuestas por submodelos que describen como se comportan los vehículos: la distancia que mantienen al auto de adelante, la velocidad de viaje, la manera en que aceleran, la ruta que escogen, como cambian de

carril, si dan vía o no, etc. Estos submodelos a su vez, en sus expresiones analíticas están compuestos por parámetros que deben ser ajustados de manera que el software se ajuste las condiciones de la zona de estudio.

En la experiencia investigativa y consultora del grupo de investigación Sirius se ha identificado la necesidad de profundizar en el estudio de la calibración de los modelos de microsimulación disponibles para prestar un mejor servicio en simulación de tráfico. Este trabajo de grado busca responder la pregunta de investigación planteada en el numeral 1.3.

1.3. Pregunta de Investigación

¿Es diferente el desempeño de las estrategias AG, SPSA y PSO en la obtención de los valores de los parámetros que calibren el modelo de microsimulación Transmodeler para la zona comprendida entre carreras 13 y 14 con calles 18 y 25 de la ciudad de Pereira, Colombia?

1.4. Objetivo General

Evaluar las estrategias: AG, SPSA y PSO en la cuanto a la calibración del modelo de microsimulación Transmodeler para la zona del parque Olaya Herrera comprendida entre carreras 13 y 14 con calles 18 y 25 de la ciudad de Pereira, Colombia.

1.4.1. Objetivos Específicos

1.4.1.1. Realizar un análisis del estado del arte en el que se evidencie las heurísticas y/o metaheurísticas que tradicionalmente se utilizan en el proceso de calibración.

1.4.1.2. Implementar los algoritmos para realizar la calibración.

- 1.4.1.3. Seleccionar los parámetros a calibrar.
- 1.4.1.4. Calibrar los parámetros por medio de los algoritmos seleccionados.
- 1.4.1.5. Modelar la red de la zona de estudio para realizar las pruebas e implementaciones.
- 1.4.1.6. Evaluar las heurísticas y/o metaheurísticas propuestas.

1.5. Justificación

La población mundial ascenderá a 8500 millones de personas para el año 2030 (Naciones Unidas, 2018). Una de las regiones donde se ha evidenciado un crecimiento acelerado en términos poblacionales y económicos es América Latina (Sosa, 2011); particularmente en Colombia la población actual es de 50.4 millones de personas, 33% más que una década atrás (DANE, 2018). Estas dinámicas han transformando ciudades intermedias en grandes centros urbanos y con ello impactado los sistemas viales, la congestión vehicular, los índices de accidentes y la contaminación principalmente. En tal sentido, temas como el transporte y la movilidad son una de las principales ocupaciones en las agendas de los estados y organismos multilaterales (Banco Mundial, 2017).

Para hacer frentes a estos retos de sostenibilidad en el transporte y la movilidad, es necesario herramientas que soporten la toma de decisiones y permitan comprender por medio de la experimentación las dinámicas urbanas propias de cada región; de tal forma que sea posible evaluar el impacto de la implementación de políticas de infraestructura, tecnología, entre otras. Actualmente, uno de los enfoques reconocidos para realizar esta tarea en la administración de las operaciones de tráfico y movilidad son los Sistemas Inteligentes de Transporte ITS, donde la microsimulación se destaca como una herramienta importante (Rodrigues, 2014).

En los modelos de microsimulación se requieren de estrategias para encontrar los valores de los parámetros que permitan que el modelo refleje el contexto local, a tal proceso se le conoce como calibración y se reconoce como uno de los elementos más importantes a realizar para un exitoso uso de esta herramienta. Para poder llevar a cabo esta tarea se requieren de algoritmos que permitan identificar el óptimo o valores próximos a este de los parámetros, que no requieran de evaluaciones explícitas de las derivadas, que converja rápidamente o en un tiempo razonable y que no crezcan con el tamaño de la red (Dowling, 2004). En tal sentido, las técnicas heurísticas y metaheurísticas son reconocidas para la tarea de calibración, puesto que permiten realizar una búsqueda inteligente y conducen a encontrar los valores de los parámetros tal que el modelo de microsimulación se adecue lo mejor posible a las características propias de la zona de estudio.

El grupo de investigación Sirius de la Universidad Tecnológica de Pereira, en su línea de investigación en ITS ha desarrollado investigaciones que tienen como objetivo mejorar la movilidad en la ciudad. De acuerdo con el estado del arte realizado, el presente estudio aporta en el área de conocimiento debido a que no se encuentran trabajos similares a nivel local; en el ámbito nacional son pocos los trabajos desarrollados y no se reporta la implementación de la técnica PSO en el país. Este trabajo es pertinente dado que profundiza en el estudio de técnicas para calibrar el software de microsimulación Transmodeler, el cual es utilizado para comprender la dinámica urbana de la ciudad de Pereira y de esta manera encaminar la toma de decisiones al mejoramiento de la calidad de vida de los ciudadanos y el crecimiento económico.

2. El problema de calibración

2.1. Revisión del estado del arte.

Este capítulo muestra los resultados de la revisión bibliográfica del problema bajo estudio. La literatura analizada fue resultado principalmente del uso de las siguientes fuentes: a) el conjunto de artículos obtenidos a partir de buscadores como Scopus, IEEE y Web of Science, por medio de las ecuaciones de búsqueda realizadas en el título, resumen y palabras claves; b) repositorios para tesis de maestría y doctorado; c) profundización de las referencias bibliográficas; d) uso de la herramienta Tree of Science o ToS, desarrollado por (Robledo, 2014). En la Figura 1 se aprecia el resultado de la ecuación de búsqueda en la herramienta ToS. Para conocer el detalle consultar el Anexo_1_Búsquedas_Indexadas.

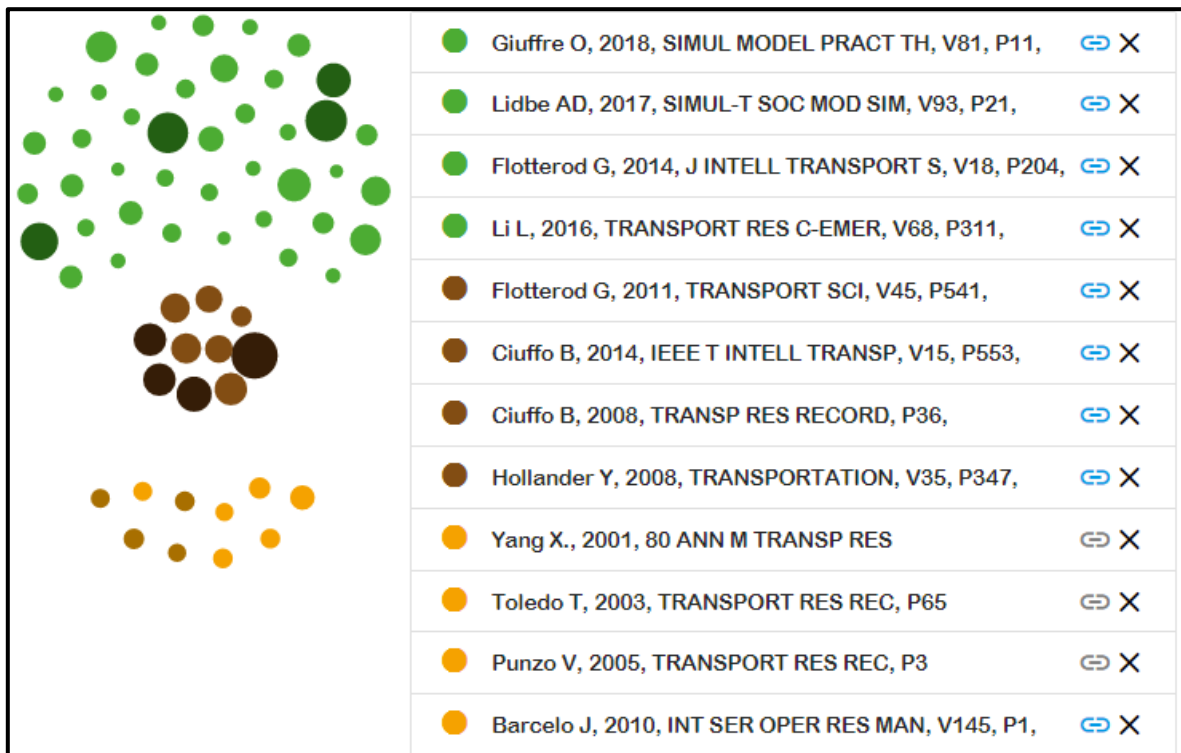


Figura 1. Árbol resultado del estado del arte en ToS. Fuente: (Robledo, 2014)

Cronológicamente la revisión se ubica desde 1998 hasta el 2017, entre las revistas consultadas se encuentran: IEEE Intelligent Transportation Systems, TRB Transportation Research Board, Journal of Algorithms and Optimization, International Journal of Transportation Science and Technology, Intelligent Information Technology Application, PATH Partners for Advanced Transit and Highways.

El estado del arte cubre la revisión de la literatura clásica de la metodología del problema de calibración, los métodos abordados para dar tratamiento y solución al problema, aplicaciones prácticas en calibración de modelos de microsimulación de tráfico, así como trabajos realizados en el país.

Un sistema tiene un enfoque holístico que se caracteriza por la interacción entre las partes para alcanzar un propósito común. En el modelamiento de un sistema se reconocen variables y parámetros, que permiten abstraer el fenómeno bajo estudio (Barceló, 2010). No obstante, en ocasiones representar todas las relaciones existentes de manera analítica es en la práctica imposible; razón por la cual se recurre a la simulación para aprender del sistema.

Los sistemas de transporte son usualmente tratados por medio de modelos de simulación de tráfico que se pueden clasificar en: macroscópicos, mesoscópicos y microscópicos. Los cuales se diferencian principalmente en el nivel de detalle utilizado para caracterizar el comportamiento del vehículo, siendo los modelos de microsimulación más detallados que los demás (Dowling, 2004).

Los modelos de microsimulación de tráfico son utilizados para evaluar diferentes estrategias que busquen mejorar la movilidad en una zona, por ejemplo, mejoras en la infraestructura, incorporación de sistemas inteligentes de control de tráfico, sistemas de información al usuario, etc (Hollander, 2008). En tal sentido, buscar evidencia a favor o en contra de la implementación estas estrategias adquieren relevancia ya que se puede evitar la

pérdida de recursos como tiempo y dinero. Estas evidencias se pueden encontrar con experimentos computacionales (Antoniou, 2014) antes de intervenir el sistema.

Para que estas evidencias sean confiables, se debe garantizar que la microsimulación represente las condiciones urbanas y patrones de conducción particulares de la zona bajo estudio; puesto que se ha observado que las características de conducción de las personas cambian entre países, incluso entre ciudades cercanas se puede apreciar un comportamiento diferente (Treiber, 2013). Esto hace que los modelos de microsimulación tengan que ser ajustados a las condiciones locales variando los parámetros que definen su comportamiento, proceso denominado calibración.

La calibración idealmente se desarrolla con datos desagregados; esto son variables medidas a cada vehículo en instantes de tiempo, por ejemplo, velocidad, aceleración, tiempos de viajes, longitudes de colas, etc. Sin embargo, obtener estos datos es infactible en la práctica; en consecuencia, datos agregados que permiten resumir la información como, por ejemplo, velocidades promedio, flujos de vehículos, etc. son utilizados por investigadores y profesionales para la calibración (Tomer Toledo, 2004).

De acuerdo con (T Toledo, 2003) los modelos de microsimulación requieren dos grupos de parámetros a calibrar como son el comportamiento del conductor y el comportamiento del viaje, más conocidos en inglés como *driving behavior* y *travel behavior parameters* respectivamente. El primer grupo controla comportamientos asociados a las condiciones particulares de conducción como por ejemplo la distancia que se deja con respecto al auto de adelante, las formas de cambio de carril o de adelantar vehículos, etc. Por su parte, el segundo grupo se ocupa de cómo los elementos del sistema eligen la ruta que les garantice un tiempo óptimo de viaje. Trabajos asociados a *travel behavior parameters* son abordados en (Tomer Toledo, 2004), (T Toledo, 2003) y (Punzo,

Vincenzo; Simonelli, 2005). Por su parte la calibración de los parámetros asociados a *driving behavior* han sido ampliamente abordados en la literatura en documentos como (Daigle, 1998), (Abdulhai, 1999) y (Punzo, Vincenzo; Simonelli, 2005). Así mismo en (Dowling, 2004) se argumenta que este grupo de parámetros afectan directamente la capacidad y deben ser calibrados en los modelos de microsimulación.

En calibración, la técnica conocida como optimización basada en simulación es utilizada para hallar el valor de los parámetros, puesto que minimiza la discrepancia entre las medidas reales y las simuladas. En consecuencia, se hace necesario una estrategia que asigne de manera inteligente valores a los parámetros, siendo las metaheurísticas un enfoque predominante (Balakrishna, 2007).

Entre las metaheurísticas más utilizadas se encuentran los algoritmos genéticos. En (T. Ma, 2001) se introduce un software piloto para calibrar de manera automática el microsimulador Paramics. En (T. Ma, 2001) los mismos autores implementan un software llamado GENOSIM compuesto por cuatro algoritmos genéticos: simple, de estado estacionario, incremental y crowding. Obteniendo como resultados valores cercanos al óptimo para algunos parámetros de los modelos *car-following* y *lane change*, utilizando como caso prueba una zona de la ciudad de Toronto, Canadá. Por otro lado, en (Wu, 2005) se implementa un algoritmo genético simple para calibrar el microsimulador VISSIM en una autopista de Shanghai, China. Los parámetros calibrados involucran la distancia deseada para cambio de carril, velocidad deseada en zonas reguladas, la distancia promedio entre dos vehículos detenidos, el tiempo que hay con respecto al automóvil de adelante a una velocidad dada y algunos parámetros del modelo de *car-following*

En (Kim, 2001) se calibran dos modelos de microsimulación, TRANSIMS y CORSIM para las ciudades de San Antonio y Houston en Estados Unidos. La calibración

fue a través de datos agregados de variables de flujo, utilizando un AG con codificación binaria. Entre los parámetros calibrados se encuentran algunos del modelo *car-following*, tasas de aceleración, distancia mínima entre vehículos, probabilidades de cambio de carril y otros. Los autores concluyen que los parámetros calibrados son mejores que los que se tienen por defecto para la zona de estudio. Finalmente, se realiza un análisis de sensibilidad de los parámetros para identificar el comportamiento en la convergencia del algoritmo genético, encontrando que no existen diferencias cuando la tasa de cruzamiento se encuentra entre 0.3 y 0.6; así mismo cuando la tasa de mutación esta entre 0.1 y 0.4. También encontraron que el tamaño de la población no afecta la convergencia después de la veinteava generación.

Los AG son una estrategia que se destaca por su sencillez en la codificación del problema, interpretabilidad y por ofrecer robustez en la búsqueda; además de haber sido probados en muchos otros campos con resultados exitosos. Sin embargo, técnicas como búsqueda tabú, partículas Swarm, búsqueda dispersa y recocido simulado han sido de igual manera reportadas en la literatura. En (Yu, 2017) se utiliza las metaheurísticas AG y búsqueda Tabú, para calibrar VISSIM en la ciudad de los Ángeles California; varios parámetros del modelo *car-following* fueron calibrados a través de Matlab.

En (Lidbe, 2017) se calibraron el modelo Wiedemann de *car-following* en VISSIM para la ciudad de Alabama, Estados Unidos utilizando las técnicas de recocido simulado, búsqueda tabú y AG. Los resultados, muestran mejores valores en términos de convergencia para la técnica de búsqueda tabú. En (Abdalhaq, 2014) se calibró el microsimulador SUMO (software libre) a través de las técnicas búsqueda tabú, AG, partículas Swarm y SPSA. En los parámetros calibrados se encuentran los modelos: *car-following* y velocidad máxima de viaje. Se concluye que el tiempo de convergencia es

menor en los algoritmos genéticos y partículas de enjambre, siendo este último el de mayor potencial.

Siguiendo esta misma línea, en (Brockfeld, 2004) se calibran diez micromodelos de *car-following* utilizando el método *Down Hill*. Se concluye que no hay un micromodelo que supere en rendimiento a los demás. Así mismo, los micromodelos con más parámetros no necesariamente representan mejor la realidad y pueden generar un sobre ajuste.

El SPSA, por sus siglas en inglés de *simultaneous perturbation stochastic approximation algorithm*, es uno de los algoritmos que más se abordan en la literatura, este algoritmo fue propuesto por James Spall (J. Spall, 2003a), se basa en una aproximación al gradiente que identifica la dirección de máximo descenso para encontrar la solución al problema de optimización sin requerir una evaluación explícita de la derivada, es decir sin conocer la relación entre los parámetros y las salidas del simulador. En (Lee, 2008) se propone una metodología de calibración para estimar los parámetros usando el algoritmo SPSA en una macrosimulación llevada a cabo en California. Esta propuesta se basa en un muestreo bayesiano, incorporando principalmente condiciones de cambio en el tiempo para una distribución de demanda. Entre tanto (J. Ma, 2007), se propone el mismo algoritmo para calibrar Paramics, los parámetros considerados son el tiempo medio de reacción, la agresividad del conductor, señales de parada, así como un factor para el cambio de carril, entre otros. Un AG es también implementado; los resultados muestran una convergencia mayor del SPSA frente AG. Así mismo, en (B Ciuffo, 2012a) se aporta una estimación de la matriz Hessiana para mejorar el SPSA.

Los algoritmos antes mencionados, también son utilizados para estimar la matriz origen destino, OD, y calibrar el modelo. En (Tomer Toledo, 2004) y (Chu, 2004) se calibra esta matriz y ciertos parámetros del modelo. La idea central consiste en estimar la

matriz OD que mejor ajuste a los flujos observados en campo, tomados en lugares particulares y en diferentes periodos de tiempo. Posteriormente un ajuste refinado a los parámetros se lleva a cabo con datos desagregados como las trayectorias, el problema se detiene cuando se llega a un nivel de convergencia.

El problema de la calibración se ve influenciado tanto por los algoritmos de búsqueda como por la función objetivo planteada. En (B Ciuffo, 2012b) se proponen cinco algoritmos: el SPSA, AG, recocido simulado, *Downhill Simplex* y *Optquest MultiStar* y catorce funciones objetivo. Los autores realizan un metamodelo a través de la técnica *kriging* para disminuir el tiempo del proceso de calibración; lo anterior es llevado a cabo para evaluar el comportamiento que puedan tener los algoritmos y la función objetivo propuesta. Se concluye que la función a optimizar influye significativamente en la estrategia de búsqueda y que el *Optquest MultiStar* encuentra el óptimo con mayor frecuencia que los demás algoritmos. Estos mismos autores, abordaron la calibración desde una perspectiva multiobjetivo que permite una búsqueda más eficiente en el espacio de soluciones. En (Huang, 2009) un algoritmo genético multiobjetivo NGSII fue abordado para calibrar el modelo en VISIM teniendo como objeto maximizar la velocidad y la capacidad de la red.

Finalmente, otra posibilidad para hallar los parámetros y realizar la calibración, son los métodos basados en juicios de expertos, en trabajos como (Huang, 2008) y (Gardes, 2002) los investigadores modifican los parámetros en unos intervalos definidos hasta que se alcanza un nivel aceptable de ajuste.

En Colombia, en (Paz, 2015) la universidad del Cauca en colaboración con la universidad de Nevada, realizan la aplicación de técnicas híbridas y se propone un algoritmo memético, como una combinación entre un algoritmo genético y uno de recocido

simulado para calibrar CORSIM. Con el primer algoritmo se explora el espacio de soluciones mientras que el segundo se utiliza para refinar la búsqueda. El algoritmo memético fue comparado con el SPSA, obteniendo mayor tiempo de simulación en el memético. Estos mismos autores en (Cobos, 2016) mejoran la investigación anterior, sustituyendo el AG por NSGA-II que es un algoritmo multiobjetivo con el que se busca minimizar la discrepancia entre el flujo y la velocidad. En esta ocasión los tiempos de cómputo fueron entre el 26% y 40% mejores para el algoritmo propuesto que para los algoritmos descritos en el trabajo anterior.

En la Universidad Tecnológica de Pereira, el grupo de investigación Sirius ha trabajado el problema de la calibración como se evidencia en (Jiménez, 2016), allí se desarrolla un software llamado Gauss para calibrar Transmodeler a través de dos AG: uno elitista y otro de *Crowding*. El presente trabajo se enmarca en esta línea de investigación y busca ampliar los algoritmos de búsqueda que utiliza Gauss.

De acuerdo con la revisión del estado del arte y la capacidad actual del grupo de investigación para recolectar información de campo, se deben descartar el enfoque multiobjetivo del presente trabajo, puesto que sólo se cuenta con variables de flujo. Este trabajo profundiza en una de las técnicas que más se aborda en la literatura como son los AG y SPSA. Tanto en el estado del arte como en Gauss la codificación utilizada para los AG es binaria y por tanto este trabajo propone explorar una codificación real. El SPSA, se hace atractivo para la investigación debido a que estima el gradiente sin una evaluación explícita de la derivada, información que debe permitir explotar un óptimo local. Por último, se implementa un algoritmo de Optimización Partículas Swarm, PSO por sus siglas en inglés, debido a la poca utilización de esta técnica en la literatura y a que su estructura debiera permitir explorar el espacio de soluciones y converger más rápido que los AG.

En la sección 3, se profundiza en el marco teórico de las técnicas seleccionadas para implementar en la presente investigación. En el Anexo_2_Estado del arte se puede consultar la información detallada de la presente sección.

3. Marco Teórico.

En el capítulo anterior, se identificaron diferentes técnicas de solución para la calibración de modelos de microsimulación. De acuerdo con el *No Free lunch theorem* propuesto en (Wolpert, 1997), no existe un algoritmo de desempeño superior en comparación con otros algoritmos para todos los problemas, y por tanto cada problema tiene un algoritmo que mejor se comporta para el caso particular. Basados en este teorema (Biagio Ciuffo, 2014), evalúan el desempeño de diferentes algoritmos para la microsimulación.

Esta investigación, selecciona e implementa un algoritmo genético de estado estacionario con codificación real. También, la técnica de perturbación simultánea para una aproximación estocástica, más conocido como SPSA y finalmente, se incorpora un algoritmo de Optimización Partículas Swarm. Los algoritmos seleccionados se consideran pertinentes para la solución del problema de calibración del modelo de microsimulación Transmodeler, para la zona de estudio del parque Olaya Herrera comprendida entre carreras 13 y 14 con calles 18 y 25 de la ciudad de Pereira, Colombia. Por lo anterior, en el presente capítulo del documento se profundiza en el marco teórico de los algoritmos seleccionados y en las técnicas estadísticas que permiten evaluar los resultados.

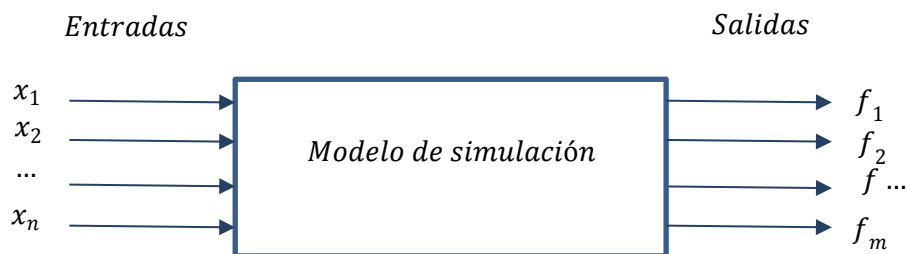
En la actualidad, el grupo de investigación Sirius no cuenta con la implementación en Transmodeler de los algoritmos seleccionados en esta investigación para trabajar el

problema de calibración. Estos son ampliamente utilizados en el estado del arte a nivel internacional y cuentan con poca acogida en el contexto nacional.

Finalmente, el problema de calibración involucra la optimización, simulación y metaheurísticas; áreas esenciales de formación académica en el programa Maestría en Investigación Operativa y Estadística, al que pertenece este trabajo, por lo que el estudio y aplicación de estos algoritmos en contextos reales es pertinente y relevante.

3.1. Optimización basada en simulación o *Simulation-Optimization*.

Un proceso de simulación puede ser representado como se muestra en Figura 2; donde se establece un conjunto de n variables de entradas obteniendo como resultado m salidas:



En la ecuación (3.1.1) se muestra que un modelo de microsimulación se puede representar como una función de variable real, cuyo dominio y codominio son vectores n y m dimensionales respectivamente; donde ε representa la variabilidad que está implícita en los modelos de simulación.

$$f: R^n \rightarrow R^m + \varepsilon \quad (3.1.1)$$

Por su parte en los modelos de optimización, generalmente las relaciones analíticas de la función objetivo y las restricciones son conocidas o se pueden establecer (Fu, 2013); Sin embargo, en algunos casos tales relaciones no se pueden determinar debido a la

complejidad del problema y no posible encontrar una solución exacta; por tal motivo es requerido un simulador para realizar el cálculo de la función objetivo f , un proceso que se conoce como optimización basada en simulación.

En la Figura 3 se puede apreciar el proceso optimización basada en simulación, el cual inicia con un conjunto variables x_1, x_2, \dots, x_n , estas son simuladas para obtener un conjunto de m observaciones f_i^s , las cuales son utilizadas en una función objetivo para medir la discrepancia a unas observaciones tomadas en campo representadas por medio de f_i^r .

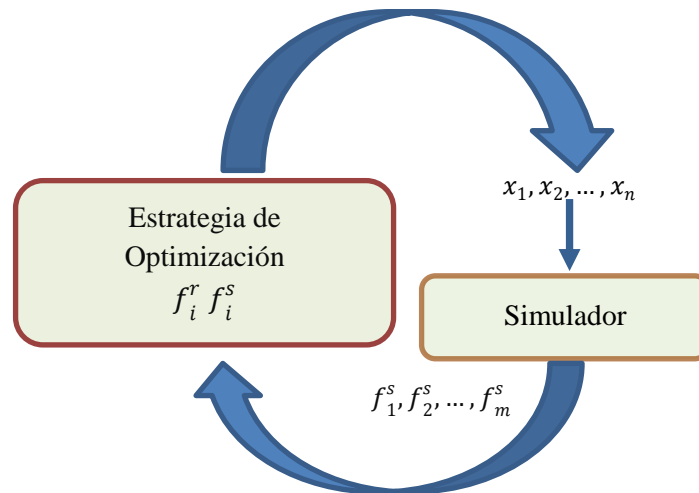


Figura 3. Optimización basada en simulación. Fuente: Elaboración Propia.

El modelo de optimización representado en la Figura 3 de acuerdo con (J. Spall, 2003a) puede ser escrito por la ecuación (3.1.2). Donde x es un vector n dimensional de parámetros a ser ajustados y $\theta \subseteq R^n$, corresponde con el dominio de x . De manera que lo que se busca es $x^* \in \theta: f(x^*) \leq f(x) \quad \forall x \in \theta$.

$$\min_{x \in \theta} f(x) \quad (3.1.2)$$

Según (B Ciuffo, 2012c), la ecuación (3.1.2) puede ser reescrita como muestra la ecuación (3.1.3) cuando se trata de un proceso de calibración; debido a que se busca la

minimización de la función objetivo en términos de las observaciones reales y simuladas. La ecuación (3.1.4), garantiza el límite superior e inferior entre los que se puede encontrar cada uno de los n parámetros.

$$\min f(MoP^s, MoP^r) \quad (3.1.3)$$

$$\begin{aligned} & \text{sa.} \\ & lb_i \leq x_i \leq Ub_i \forall i \in n \quad (3.1.4) \end{aligned}$$

3.2. Proceso de calibración.

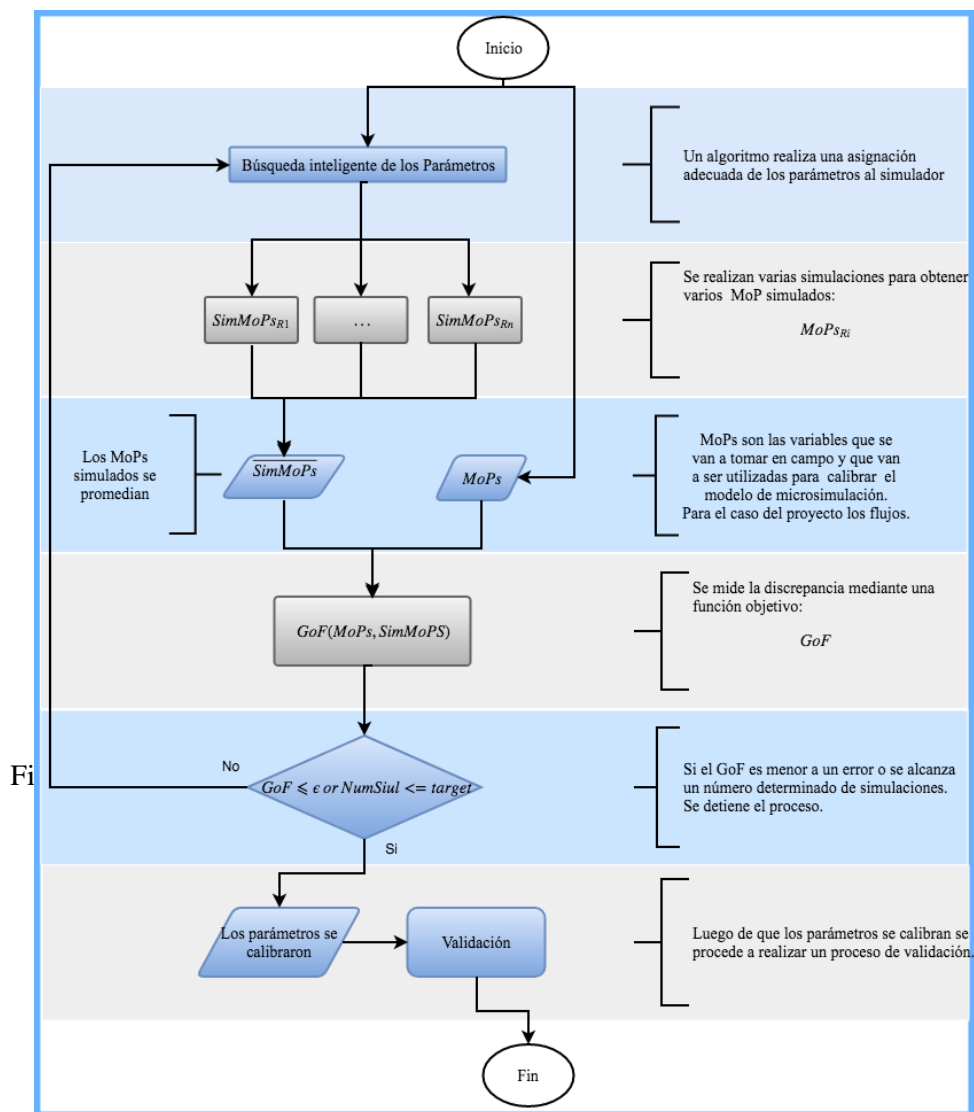
En la calibración de los modelos de microsimulación se busca encontrar los valores de los parámetros que mejor ajusten el modelo a la zona de estudio. Lo anterior se debe a que los microsimuladores tienen un conjunto de submodelos que caracterizan diferentes comportamientos tales como: la manera en que aceleran los vehículos, la elección de la ruta, la velocidad deseada de viaje, distancia al auto de adelante, distancia al auto de atrás para cambiar de carril etc. En las expresiones analíticas de la mayoría de estos submodelos se tiene la presencia de parámetros que modifican el comportamiento de los vehículos de acuerdo con los valores asignados a estos y por ende el de la simulación.

En la Figura 4 se detalla el proceso de calibración. La nomenclatura utilizada es:

$MoPs$	Medidas de rendimiento reales de la red (velocidades, flujos, tiempos de viaje, etc.).
$SimMoP_{Ri}$	Medidas de rendimiento simuladas de la red, en la réplica i – ésima.
\overline{SimMoP}	Medidas promedio de rendimiento simuladas de la red.
GoF	Función Objetivo que minimiza la discrepancia entre $SimMoP$ y $MoPs$.
ε	Error de discrepancia admisible
$Target$	Parámetro de la cantidad máxima de simulaciones a realizar.
$NumSiul$	Simulación i –ésima en la corrida.

La Figura 4 muestra el proceso de calibración, este inicia con una búsqueda inteligente de los parámetros realizada por medio de heurísticas y/o metaheurísticas. Posteriormente, se realiza n réplicas con los parámetros seleccionados por el algoritmo, para obtener así los valores de las medidas de rendimiento simuladas $SimMoP_{Ri}$, y con obtener con ellas un promedio de estas simulaciones; estos valores son comparados con las medidas de rendimiento reales a través de función objetivo GoF , que estima la discrepancia entre estas. El proceso se repite hasta que se cumple un criterio de parada, que puede ser un máximo de iteraciones o que el valor de la función objetivo sea menor o igual a ε , la finalización del proceso se traduce en que el algoritmo encontró los parámetros que

calibran el modelo de microsimulación. Posteriormente, un proceso de validación debe ser llevado a cabo donde se verifica si los parámetros calibrados presentan un mejor ajuste a la realidad bajo estudio en comparación con los parámetros por defecto del software.



3.3. Algoritmo de optimización.

Dentro del proceso de calibración descrito en la Figura 4 se identifica una etapa de búsqueda inteligente de los parámetros para llevar a cabo el proceso de optimización basada en simulación. La calibración puede realizarse a través de varias técnicas como las señaladas en el estado del arte; a continuación, se describen las tres que se implementan en la presente investigación.

3.3.1. Partícula Swarm.

La metaheurística Particula Swarm Optimization o PSO, es un caso particular de las metaheurísticas basadas en inteligencia de enjambre, que están inspiradas en el comportamiento social de ciertos animales como las hormigas, aves, peces y abejas. Estas sociedades están compuestas por individuos que realizan actividades simples y que de manera individual no sobrevivirían, por lo que obtienen ventaja al tomar información del entorno y compartirla (Marini, 2015). El más conocido de este tipo de algoritmo es el llamado colonia de hormigas, este es utilizado para resolver problemas que puedan ser expresados en forma de grafos.

De acuerdo con (Xinjie, 2010) la optimización mediante enjambre de partículas fue introducida por Kennedy y Eberhart en 1995. Esta permite resolver problemas cuyo espacio de soluciones o dominio es multidimensional y las partículas se pueden mover en él libremente teniendo en cuenta su mejor posición, la mejor posición encontrada por un individuo en el enjambre y la velocidad que implícitamente lleva consigo. De acuerdo con (Marini, 2015), en general se tiene la función expresada por la ecuación (3.3.1).

$$f: R^n \rightarrow R \quad (3.3.1)$$

La ecuación (3.3.2) representa los valores óptimos del vector de parámetros que el investigador decide calibrar como x_a ; este vector es simulado de tal manera que se estima $f(x_a)$ tal que debe ser mínima en el dominio de x .

$$x_a \forall f(x_a) \leq f(x) \forall x \in R^n \quad (3.3.2)$$

Según (Marini, 2015), en PSO existe una población de i partículas dispersas en el espacio de soluciones de tal manera que el individuo i –ésimo puede ser representado por la ecuación (3.3.3), donde n es la dimensión del espacio de soluciones determinada por la cantidad de parámetros a calibrar.

$$x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\} \quad (3.3.3)$$

Las posiciones de cada una de las partículas pueden ser actualizadas mediante la ecuación (3.3.4) como bien se describe en (Marini, 2015):

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3.3.4)$$

Donde t representa la iteración t –ésima y v_i representa la velocidad de la partícula i –ésima la cual es estimada por la ecuación (3.3.5).

$$v_i(t+1) = v_i(t) + c_1(p_i - x_i(t))R_1 + c_2(G - x_i(t))R_2 \quad (3.3.5)$$

El primer término de la ecuación (3.3.5), $v_i(t)$, es el momento de inercia de la partícula que impide que esta cambie de posición de manera drástica. El segundo término, $p_i - x_i(t)$ es llamado componente cognitivo, permite que la entidad vuelva desde su posición aleatoria a la mejor encontrada hasta el momento. Por último $G - x_i(t)$ dirige al individuo a la mejor posición encontrada por todo el enjambre y representa el componente social.

c_1 y c_2 son los coeficientes de aceleración que controlan la magnitud del movimiento de la partícula, hacia su mejor posición y la encontrada por el enjambre respectivamente. Por su parte, R_1 y R_2 son dos matrices diagonales de números aleatorios

uniformes entre 0 y 1 los cuales generan un comportamiento estocástico de la velocidad. Finalmente p_i y G representan la mejor solución encontrada por la entidad i –ésima y el enjambre. Al modelo de las ecuaciones anteriores se les conoce como PSO orgánicos (Xinjie, 2010).

Como la gran mayoría de metaheurísticas, PSO requiere del ajuste de los coeficientes para un correcto funcionamiento. En la literatura, los individuos iniciales se generan de manera uniforme en el espacio de búsqueda. Como bien puede notarse en la (3.3.5) cuando $t = 0$ la mejor posición de la partícula es igual a su estado inicial por tanto el componente cognitivo se hace igual a cero $v_i(t = 0) = 0$, lo que lleva a que el componente social domine la velocidad en el inicio del algoritmo y en consecuencia puede llevar a una convergencia local prematura. Por tal motivo se sugiere que la velocidad inicial sea generada uniformemente en el espacio de búsqueda.

Así mismo, cuando $t = 0$ el momento de inercia domina el comportamiento, de manera que cuando se actualiza la posición mediante la ecuación (3.3.4) se experimenta un cambio abrupto en la posición algo que se conoce como explosión de velocidad, lo anterior ocasiona que las partículas se alejen de los límites del espacio de soluciones. Para evitar esto, en (Marini, 2015) se proponen diferentes estrategias que impidan una oscilación en las partículas y que estas vuelen al infinito. Una de estas consiste en definir un límite para la velocidad, lo difícil de esta decisión es: ¿cómo se debe escoger este límite de velocidad? los autores proponen la ecuación (3.3.6) que formula una velocidad máxima dada por la siguiente expresión:

$$v_n = \frac{x_{n,max} - x_{n,min}}{2} * k \quad (3.3.6)$$

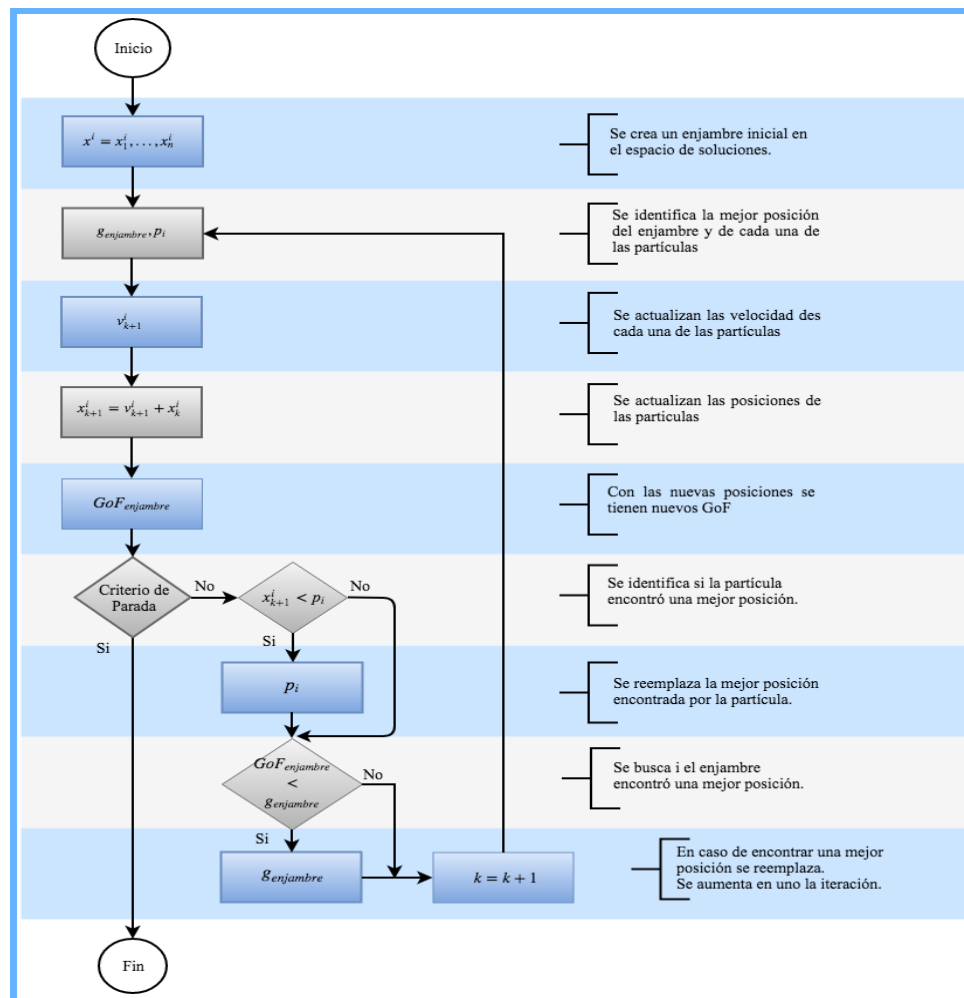
Donde v_n representa el límite máximo de velocidad para la dimensión n -ésima del espacio de soluciones, $x_{n,max}$ es el límite superior del n -ésimo parámetro y $x_{n,min}$ el límite inferior; k es una constante en el intervalo.

En (Xinjie, 2010), se incorpora un nuevo término para realizar la actualización de la velocidad llamado coeficiente de inercia de manera que la ecuación (3.3.5) puede reescribirse como se presenta en la ecuación (3.3.7). De tal forma que si $w > 1$ favorece la exploración, mientras que $w < 1$ predomina la búsqueda local.

$$v_i(t + 1) = wv_i(t) + c_1(p_i - x_i(t))R_1 + c_2(G - x_i(t))R_2 \quad (3.3.7)$$

w representa el término de inercia que puede cambiar de forma dinámica como se evidencia en (Marini, 2015), permitiendo que al inicio se promueva una búsqueda global y luego una local, como se muestra en la ecuación (3.3.8).

$$w(t) = w_{max} - \frac{w_{max} - w_{min}}{t_{max}} t \quad (3.3.8)$$



Nótese en la Figura 5. Estructura del PSO. Fuente: Elaboración Propia. en la ecuación (3.3.8) que cuando se alcanza el mayor número de iteraciones dado por t_{max} , la inercia va a ser igual al valor mínimo. En síntesis, la Figura 5 se ilustra y expone los pasos para implementar un algoritmo de enjambre de partículas.

3.3.2. SPSA: Simultaneous Perturbation Stochastic Approximation

En análisis matemático el cálculo del gradiente permite identificar la dirección en que la función crece o decrece más rápidamente. Este algoritmo busca aproximar el gradiente sin necesidad de conocer explícitamente la función objetivo. Esta es una característica importante, debido a que uno de los principales inconvenientes en calibración es que dicha función no es conocida. Esta metodología se basa únicamente en las medidas de la función objetivo, empleando un razonamiento similar al método de diferencias finitas. El modelo recursivo general puede ser apreciado en la siguiente ecuación (J. Spall, 2003a).

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) \quad (3.4.1)$$

La actualización de la posición hacia el valor mínimo de la función objetivo está dada por la ecuación (3.4.1), donde $\hat{\theta}_k$ representa la estimación para el vector óptimo θ^* en la iteración k . Ambos vectores son de dimensión p determinados por la cantidad de parámetros a calibrar. El término $\hat{g}_k(\hat{\theta}_k)$ es la estimación del gradiente, esencia del método. El valor a_k es un factor de ganancia que debe ser no negativo debido a que se está minimizando una función objetivo.

El gradiente $\hat{g}_k(\hat{\theta}_k)$ de ecuación (3.4.1) es estimado con las medidas de la función objetivo, de acuerdo con la ecuación (3.4.2) (J. Spall, 2003a).

$$\hat{g}_k(\hat{\theta}_k) = \begin{array}{c} \frac{y(\hat{\theta}_k + c_k \Delta_k) - y(\hat{\theta}_k - c_k \Delta_k)}{2c_k \Delta_{k1}} \\ \vdots \\ \frac{y(\hat{\theta}_k + c_k \Delta_k) - y(\hat{\theta}_k - c_k \Delta_k)}{2c_k \Delta_{kp}} \end{array} \quad (3.4.2)$$

En la ecuación (3.4.2) los numeradores son iguales y corresponden con la diferencia de dos medidas de la función objetivo, obtenidas al perturbar el vector de parámetros en una cantidad igual a $c_k \Delta_k$. Los denominadores; por otro lado, cambian de acuerdo con la perturbación realizada al parámetro, por lo que el cociente da una idea de la dirección de descenso más rápida.

Existen diferentes métodos que aproximan el gradiente entre los que sobresale el FDSA, por sus siglas en inglés de *Finite Differences Stochastic Approximation*, y el SPSA. Este último algoritmo, solo requiere dos mediciones de la función objetivo, mientras que el FDSA requiere tantas medidas como parámetros tenga el problema lo que hace perder potencial en términos de tiempo. En consecuencia, el SPSA es el método de perturbación estocástica más empleado en la actualidad (J. Spall, 2003a).

Como todos los algoritmos, el SPSA requiere algunas consideraciones para garantizar convergencia (J. Spall, 2003a). Las secuencias de ganancia de los coeficientes a_k y c_k se calculan por medio de las ecuaciones (3.4.3) y (3.4.4) respectivamente. Estos coeficientes deben escogerse de manera que decrezcan conforme aumente el número de iteraciones. En (J. C. Spall, 1998) establecen los valores para Alpha y Gamma más pequeños que permiten satisfacer las consideraciones teóricas. Estos son 0.602 y 0.101 respectivamente.

$$a_k = \frac{a}{(A + k + 1)^\alpha} \quad (3.4.3)$$

$$c_k = \frac{c}{(k + 1)^\gamma} \quad (3.4.4)$$

Este mismo autor argumenta que cuando el valor de la función objetivo está sometida a la presencia de mucho ruido, producto de que el proceso es estocástico, se recomienda un valor grande de c y uno pequeño para a , experimentalmente 0.16. Para la primera de ellas se aconseja utilizar la desviación estándar de la función en un valor inicial, $y(\hat{\theta}_0)$, con el objetivo de obtener elementos grandes en el gradiente para garantizar una diferencia considerable. Así mismo, A debe ser mucho menor que la cantidad total de iteraciones, generalmente se toma un valor menor al 10%. Por último y no menos importante, la perturbación que se da cada uno de los elementos en θ deben ser variables aleatorias independientes, la más utilizada es una distribución de Bernulli.

En síntesis, en la Figura 6 se detallan y sintetizan los pasos para implementar el algoritmo SPSA.

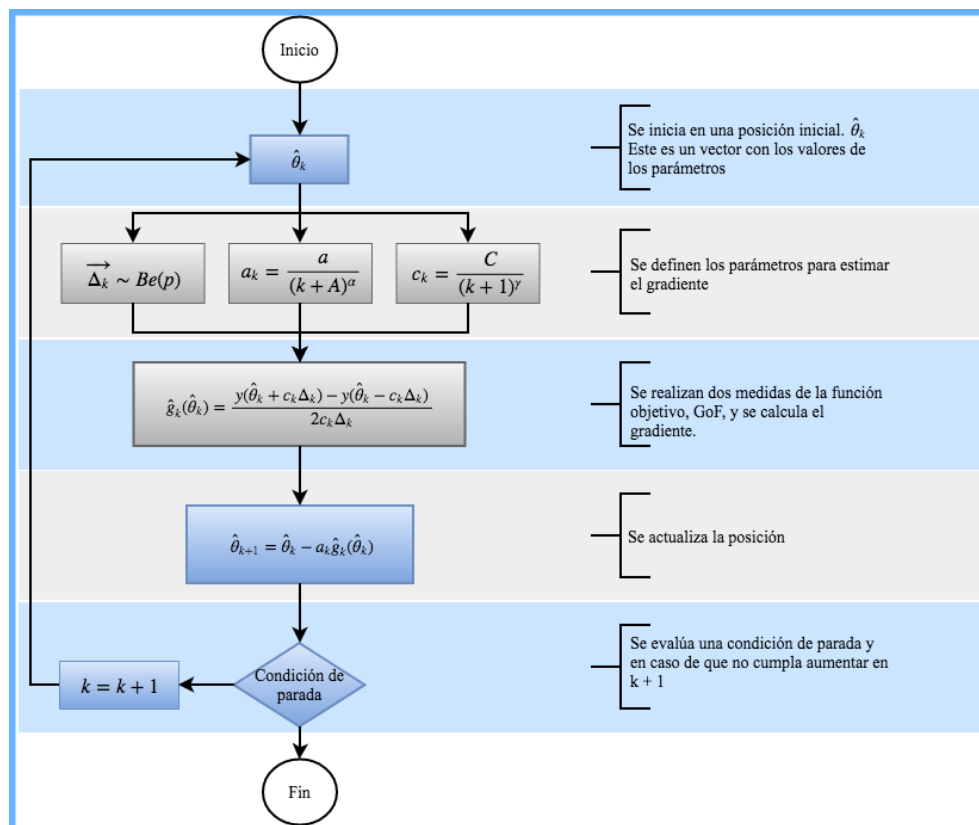


Figura 6. Estructura del SPSA. Fuente: Elaboración Propia.

3.3.3. Algoritmo genético.

La investigación de operaciones enfrenta diversos tipos de problemas como, por ejemplo: combinatoriales, multimodales, optimización restringida, multiobjetivos entre otros (Xinjie, 2010); estos tienen en común que no es posible encontrar una solución basada en gradiente. Por ejemplo, si la función es multimodal, la respuesta dependerá del punto de inicio; si existen restricciones, el gradiente puede encontrar una solución que es infactible al problema y en muchos otros no se cuenta la información de la función objetivo de manera explícita. Se requiere entonces de otras estrategias que permitan solventar estas dificultades.

La computación evolutiva se erige como una de las tecnologías que permite abordar las problemáticas anteriores. Entre estas, se destacan los algoritmos genéticos. Estos fueron introducidos en (Holland, 1992). Esta metaheurística busca imitar el proceso de selección natural en la cual los individuos mejor adaptados tienen mayores probabilidades de sobrevivir y reproducirse.

Según (Eiben, 2008) el fenómeno evolutivo se presenta en la naturaleza debido a que los individuos que componen una población tienen dos características: un genotipo y un fenotipo. La primera de ellas hace referencia a la información genética de cada una de las entidades; mientras la segunda, es la manifestación física y/o cognitiva de la primera, y permiten a los organismos de cada especie generar ventajas o desventajas en su medio ambiente. Los genotipos se ven enriquecidos por mezclas en la información cromosómica gracias a un instinto de reproducción presente en las especies. En ocasiones, durante el proceso de cruzamiento y generación de un nuevo individuo ocurren mutaciones, errores de la naturaleza que ocurren de manera aleatoria y que cambian la información contenida en los genes y por consiguiente su fisonomía. Si la manifestación de la mutación es positiva, el

organismo experimenta una nueva cualidad que lo hace fuerte al medio ambiente y sus probabilidades de sobrevivir y tener descendencia aumentan, llevando la nueva información a sus descendientes. Por otro lado, sí la manifestación de la mutación es negativa es una desventaja y lo más probable es que perezca, tal que la nueva información no se transmita en futuras generaciones. Lo anterior ocurre de manera que se depura la población dando origen a organismos mejor adaptados.

Los algoritmos genéticos emulan el fenómeno anteriormente descrito para resolver diversos tipos de problema. En la Tabla 1 se observa un paralelo entre algunas características evolutivas, un problema de optimización y un ejemplo. Permitiendo identificar como se lleva a cabo la abstracción de las anteriores ideas.

Tabla 1. Evolución vs. Optimización.

Evolución	Investigación de Operaciones	Ejemplo
Medio Ambiente	Problema de Optimización	$\min f(x,y) = \frac{x^2 + y^2}{4000} - \cos(x)\cos\left(\frac{y}{\sqrt{2}}\right) + 1$ $-30 \leq x, y \leq 30$
Organismo	Solución	$(x, y) = [-5.324, 10.31]$
Adaptación	Valor de la función Objetivo.	0.7268569

Adaptada de : (Eiben, 2008).

La función y la solución propuesta como organismo mostrado en el ejemplo de la Tabla 1 se ilustra en la Figura 7.

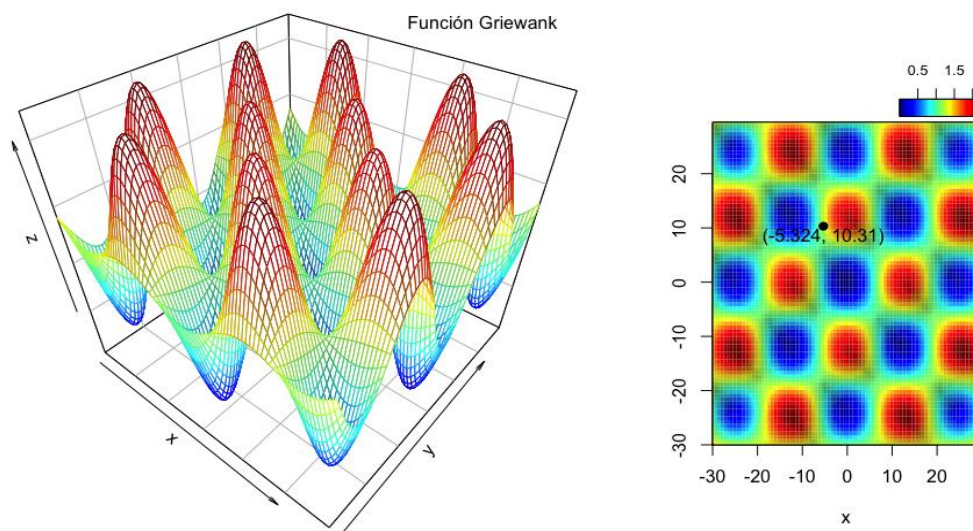


Figura 7. Función Griewank. Adaptada de: (Eiben, 2008).

3.3.3.1. Modelos de algoritmos genéticos.

En el campo de los algoritmos genéticos se pueden distinguir entre dos modelos: generacionales y de estado estacionario (Eiben, 2008). Para los primeros se seleccionan un conjunto de individuos, estos luego son cruzados y mutados para generar tantos hijos como individuos iniciales para luego reemplazar a los primeros y dar origen a una nueva generación. En los modelos estacionarios la población no se cambia toda de una sola vez sino una parte de ella; tal que se crean una cantidad de individuos menores a la población que reemplazan a los individuos peor adaptados.

3.3.3.2. Etapas de en el proceso de los algoritmos genéticos.

Los pasos que llevan a cabo los algoritmos genéticos de manera general se exponen en la Figura 8:

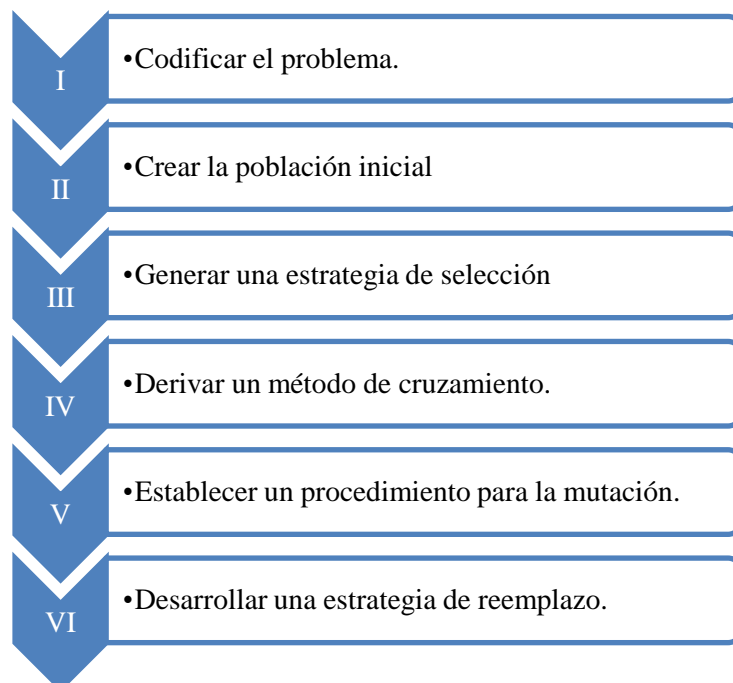


Figura 8. Pasos para implementar AG. Fuente: Elaboración propia.

3.3.3.3. Codificación Real.

Los algoritmos genéticos inician con un conjunto de individuos que representan soluciones al problema. Por tal motivo uno de los primeros pasos cuando se emplea esta estrategia es realizar una codificación. Durante largo tiempo la codificación binaria fue la más utilizada debido a que con base a esta se desarrollaron todas las propiedades teóricas y de convergencia de estos algoritmos, teoría de esquemas (J. Spall, 2003b). Así mismo, esta codificación permite la interpretación de los operadores de cruzamiento y mutación de manera sencilla.

Por otro lado, una codificación real permite representar muchos problemas de ingeniería de manera más natural; además otorga ventajas como: no requiere codificar y decodificar, la precisión no está asociada con la longitud del cromosoma sino con la cantidad de bits del computador. La codificación real tomó fuerza a finales de la década de los ochentas (Whitley, 1988) y a inicios de los noventas se ahondaron en los esfuerzos para llegar a una teoría de convergencia con esta representación tal como se expone en (Goldberg, 1991).

Un individuo X puede representarse por medio de una codificación real como un vector n dimensional con tantas columnas como variables tenga el problema, como se muestra en la ecuación (3.5.1), de tal manera que x_i representa la i -ésima componente del individuo X . Además, la ecuación (3.5.2) hace referencia los límites máximos ub_i y mínimos lb_i que pueden tomar cada componente.

$$X = x_1, x_2, \dots, x_n \quad (3.5.1)$$

$$lb_i \leq x_i \leq ub_i \quad (3.5.2)$$

3.3.3.4. Población.

La población representa el conjunto de individuos que van a ser sometidos a los procesos evolutivos. Usualmente esta no cambia a medida que aumentan las generaciones y se busca sea lo más heterogénea posible, es decir rica en diversidad genética. Para lograr lo anterior esta se crea distribuida uniformemente sobre el espacio de soluciones.

3.3.3.5. Estrategia de Selección.

Existen diferentes estrategias para seleccionar cuales van a ser los padres que van a ser sometidos al fenómeno de cruzamiento. Entre las que se destaca la selección por torneo, por su fácil implementación y conceptualización. Esta selección depende de tres factores: tamaño de torneo k , son la cantidad de individuos a seleccionar de la población; otro factor es la probabilidad con la que se va a seleccionar el mejor individuo de la muestra k , cuando esta es igual a 1, se trata de un torneo determinístico, de lo contrario es un torneo estocástico. Finalmente, el tercer factor se asocia a identificar si la selección es con o sin reemplazo (Eiben, 2008).

3.3.3.6. Estrategia de cruzamiento.

El cruzamiento busca enriquecer en la población el material genético, al intercambiar información entre dos padres. Existen varios tipos de cruzamiento en la literatura algunos de ellos son: aritmético, lineal, extendido, convexo y plano (Eiben, 2008). Particularmente, el cruzamiento plano genera al descendiente mediante un número uniforme entre los alelos de cada uno de los padres. Por ejemplo, si se asume que el padre uno está dado por la ecuación (3.5.3), el padre dos por la ecuación (3.5.4) y que x_{ij} representa el valor de la variable j -ésima del padre i -ésimo.

$$P_1 = x_{11}, x_{12}, \dots, x_{1n} \quad (3.5.3)$$

$$P_2 = x_{21}, x_{22}, \dots, x_{2n} \quad (3.5.4)$$

Se tiene entonces que el hijo representado por la ecuación (3.5.5) es generado de tal manera que cada h_s esta dado por la ecuación (3.5.6).

$$H = h_1, h_2, \dots, h_n \quad (3.5.5)$$

$$h_s \sim U(x_{ks}, x_{ws}) \quad (3.5.6)$$

Vale la pena señalar que cada componente del hijo se elige de forma aleatoria y uniforme entre los valores de los padres lo que permite acotar el espacio de soluciones a medida que la calidad de la población aumenta con las generaciones.

3.3.3.7. Estrategia de Mutación.

Una de las maneras más sencillas de realizar la mutación para la codificación real es realizarla de manera uniforme (Herrera, 1998). Para esto, se selecciona un gen del cromosoma de manera aleatoria y se genera un número uniformemente distribuido entre el límite inferior y el límite superior de la variable seleccionada. La mutación se genera de manera uniforme entre el dominio de cada componente. Tal que sí permitimos que c_i represente el gen i – ésimo de un individuo cualquiera. La mutación se puede generar entonces como expresa la ecuación (3.5.7).

$$c_i \sim U(lb_i, up_i) \quad (3.5.7)$$

3.3.3.8. Estrategia de reemplazo.

Los individuos que presenten mejor función de adaptación, esto es mejor función objetivo y menor penalización por el no cumplimiento de las restricciones, son incorporados en la población de manera que reemplazan los individuos de peores calidades (Eiben, 2008).

Los pasos desde el 3.5.2.1 hasta el actual son repetitivos hasta cumplir un criterio de parada en el algoritmo, frecuentemente establecido por el investigador de acuerdo con un número de iteraciones máximos o el tiempo computacional.

Finalmente, en la Figura 9 se resumen de manera general la estructura de un algoritmo genético de estado estacionario con dos torneos en la selección y de codificación real, exponiendo las actividades en cada paso.

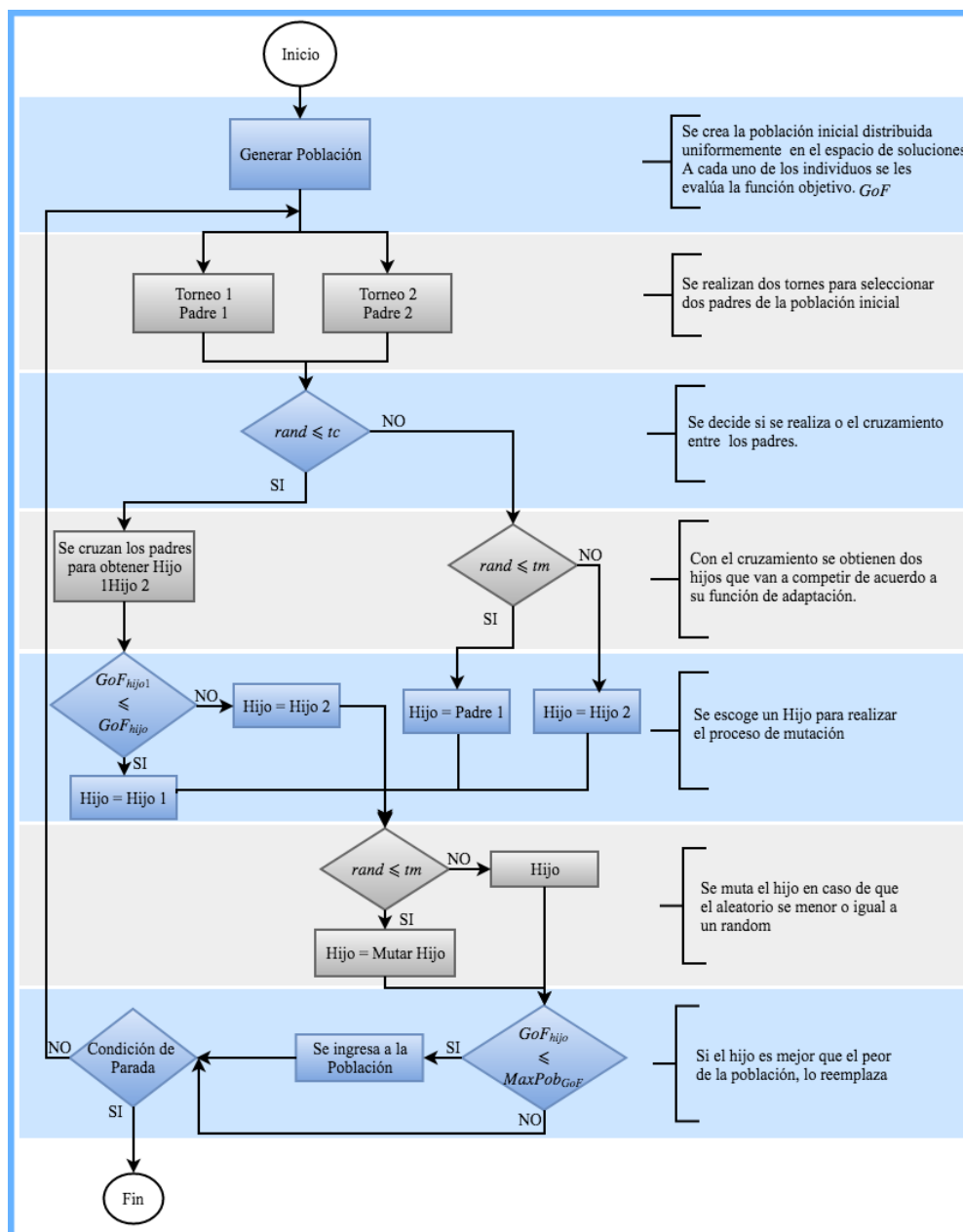


Figura 9. Estructura del AG. Fuente: Elaboración Propia.

3.4. Marco teórico para la realización de las pruebas estadísticas.

Para evaluar los resultados de los parámetros calibrados a través de los algoritmos con respecto a los valores por defecto del microsimulador, se hace necesario realizar pruebas estadísticas debido a la naturaleza aleatoria del proceso. Lo anterior es para identificar si las diferencias en la función objetivo son debido al proceso estocástico del simulador o a los resultados de la investigación (Daamen, 2014b). De igual manera se requiere identificar si el comportamiento en convergencia de los algoritmos. Para ello se requiere de los siguientes conceptos:

3.4.1. Teorema del límite central.

Para lograr la estimación de esta discrepancia, es preciso realizar una diferencia de medias, por medio de un intervalo de confianza que es soportado en el teorema del límite central.

El teorema del límite central es uno de los más importantes de la estadística. A partir de este teorema se deduce que la distribución de las medias muestrales es normal, independiente de la distribución subyacente de los datos.

Formalmente se presenta como variables aleatorias independientes e idénticamente distribuidas X_1, X_2, \dots, X_n , cuyos valores esperado y varianza están dados por las ecuaciones (3.6.1) y (3.6.2) respectivamente.

$$E(X_i) = \mu \quad (3.6.1)$$

$$V(X_i) = \sigma^2 \quad (3.6.2)$$

De tal manera que al definir la variable U como se muestra en la ecuación (3.6.3), se obtiene que U sigue una distribución normal estándar, $U \sim N(0,1)$ cuando $n \rightarrow \infty$.

$$U = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \quad (3.6.3)$$

Las ecuaciones presentadas en esta sección fueron tomadas de (Wackerly, 2009).

3.4.2. Estimadores.

En la práctica es costoso y muchas veces imposible obtener los valores de los parámetros de la población. Por ello, se recurren a muestras para estimarlos por medio de reglas o fórmulas que permiten encontrar un número que se aproxima al verdadero valor subyacente a la población, este proceso se realiza a través de los estimadores. Una de las propiedades deseadas de estos es que sean insesgados como se muestra en la ecuación (3.7.1), en otras palabras, que el valor esperado de la estimador sea igual al parámetro.

$$E(\hat{\theta}) = \theta \quad (3.7.1)$$

Generalmente, en los procesos investigativos se debe distinguir si las diferencias entre conjuntos de observaciones que provienen de diferentes experimentos son debidas a la aleatoriedad del fenómeno o a las particularidades de cada experimento. Una idea intuitiva para hacer esto es comparar la diferencia entre las medias de los experimentos \bar{X}_1 y \bar{X}_2 , tal que se puede probar la ecuación (3.7.2).

$$E(\bar{X}_1 - \bar{X}_2) = \mu_1 - \mu_2 \quad (3.7.2)$$

Así, se tiene que la diferencia de medias es un estimador insesgado para las diferencias poblacionales además de ser normales. El resultado anterior y el teorema del límite central permite crear un intervalo de confianza y se pueda establecer la probabilidad de que diferencia se encuentre en un intervalo estimado.

Las ecuaciones presentadas en esta sección fueron tomadas de (Wackerly, 2009).

3.4.3. Intervalos de Confianza.

Los intervalos de confianza permiten crear un rango en el que se puede encontrar el verdadero valor del parámetro con una probabilidad $1 - \alpha$. Si se representa un estimador como $\hat{\theta}$ se puede realizar la transformación dada por la ecuación (3.8.1) y de acuerdo con el teorema del límite central se sabe que $Z \sim N(0,1)$.

$$Z = \frac{\hat{\theta} - \theta}{\sigma_{\hat{\theta}}} \quad (3.8.1)$$

Lo anterior permite expresar la probabilidad del intervalo para la variable Z, como se muestra en la ecuación (3.8.2).

$$P(-Z_{\alpha/2} \leq Z \leq Z_{\alpha/2}) = 1 - \alpha \quad (3.8.2)$$

Al reemplazar el valor de Z en la ecuación (3.8.2), es posible hallar un intervalo para el parámetro poblacional θ , como se expone en la ecuación (3.8.3).

$$P(\hat{\theta} - Z_{\alpha/2}\sigma_{\hat{\theta}} \leq \theta \leq \hat{\theta} + Z_{\alpha/2}\sigma_{\hat{\theta}}) = 1 - \alpha \quad (3.8.3)$$

Las ecuaciones presentadas en esta sección fueron tomadas de (Wackerly, 2009).

3.5. Tamaño de la muestra.

En cualquier estudio estadístico el tamaño de la muestra juega un papel preponderante. De la muestra se realizan estimaciones que tiene un error con respecto al verdadero valor que se encuentra en la población; por esto se busca que el error sea lo menor posible y se sabe que entre más grande sea la muestra, más pequeño es el error.

Para determinar el tamaño de la muestra se debe de identificar primero el error ϵ que está dispuesto a asumir el investigador y segundo, se debe definir la probabilidad de que ocurra el error. En la ecuación (3.9.1) se define el tamaño de la muestra para realizar la estimación en términos de la desviación del estimador.

$$Z_{\alpha/2}\sigma_{\hat{\theta}} = \varepsilon \quad (3.9.1)$$

Las ecuaciones presentadas en esta sección fueron tomadas de (Wackerly, 2009).

4. Metodología propuesta

A continuación, se describen los elementos desarrollados para dar cumplimiento a los objetivos planteados en la investigación. Se inicia con la delimitación y descripción de la zona de estudio, seguidamente se expone la recolección de los datos en la información utilizada. Así mismo, se muestra la red implementada con la información de campo, las características de la infraestructura vial y señalizaciones presentes en el área de estudio.

Posteriormente, se seleccionan los modelos, parámetros y los algoritmos utilizados para la solución del problema planteado en Transmodeler, utilizando Gauss como herramienta para la calibración. También, se muestra la función objetivo que minimiza la discrepancia entre los valores simulados y observados, éstos últimos a través de los flujos reales, más conocidos en la literatura como medida de rendimiento MoP.

4.1. Delimitación y descripción de la zona de estudio.

La zona de estudio en la presente investigación comprende las carreras 13 y 14 entre calles 18 y 23 de la ciudad de Pereira Colombia. Esta se caracteriza por albergar la gobernación del departamento de Risaralda. Adicionalmente, la calle 21 es una de las principales vías urbanas que conectan el sur con el centro de la ciudad. Las carreras 13 y 14, más conocidas como la Avenida 30 de Agosto conecta al oriente con el occidente y viceversa; además es uno de los accesos fundamentales al aeropuerto de la ciudad; de igual manera esta avenida conecta con la Avenida del Ferrocarril la cual es uno de los tres puntos

de acceso que tiene la ciudad al norte del departamento y que conecta directamente con los municipios de Dosquebradas y Santa Rosa de Cabal, salida vial al norte de país.

La carrera 14 se bifurca, entre sus ramificaciones se encuentra la calle 17 que permite el acceso al terminal de transporte de Pereira y desemboca en la calle 18 que termina en la ruta nacional 29 hacía el sur del país, más conocida como la utopista del café que se extiende por los principales municipios del eje cafetero: Pereira, Armenia y Manizales.

Finalmente, la zona seleccionada se caracteriza por ser un lugar en la ciudad con amplias zonas verdes y deportivas donde cientos ciudadanos convergen para disfrutar de este lugar en actividades de esparcimiento y físicas al aire libre. Proyectos como el cable aéreo línea parque Olaya-Terminal-UTP-Villa Santa fueron puestos en marcha (Alcaldía Pareira, 2017). Este proyecto iniciará procesos de construcción y estará listo para marzo del año 2020. Las razones anteriormente expuestas hacen de la zona bajo estudio, un lugar con gran movimiento peatonal y vehicular, así como de gran importancia e interés en el desarrollo de la ciudad. En la Figura 10 se muestra la zona bajo estudio en el presente



Figura 10. Zona bajo estudio. Fuente: open street map.

trabajo.

4.2. Recolección y tratamiento de la información utilizada.

La información recolectada para realizar la calibración del modelo de microsimulación se muestra en la Tabla 2. Se consideró la estimación de la matriz origen destino, los tiempos de las fases de los semáforos, la clasificación de la flota, los flujos de los vehículos en puntos determinados y la caracterización de la infraestructura vial, presente en la zona de estudio.

Tabla 2. Resumen de la información utilizada.

Tipo de información	Descripción de la información
La matriz origen destino:	<p>Contiene la información de los flujos vehiculares que transitan desde un punto origen a los posibles destinos, en la zona de estudio.</p> <p>Para obtener esta información se ubicaron cámaras en lugares estratégicos que permitieron rastrear los vehículos, previo permiso de las autoridades competentes.</p>
Tiempos de las fases de los semáforos:	<p>Debido a que en la zona analizada existen intersecciones semaforizadas, se requiere del tiempo que dura cada una de las luces en estos dispositivos, para poder ser implementados en el microsimulador. Esta información se debió gestionar con la empresa encargada de manejar los controladores de los semáforos en la ciudad.</p>
Características de la infraestructura	<p>Corresponde con la identificación de elementos particulares del corredor vial investigado, por ejemplo, número de carriles en la vía,</p>

vial: señales de tránsito, cruces peatonales, entre otros. Esta información fue levantada en campo por una analista de transporte.

Fuente: Grupo de investigación Sirius.

La información recolectada es procesada por analistas y entre las actividades asociadas se encuentra: conteo de vehículos, depuración de los datos, digitalización, observación directa y de material multimedia.

El periodo de la observación para el levantamiento de los datos cubrió los horarios entre las 11:30 a.m. y 2:45 p.m. durante los días hábiles de una semana sin festivos, ni eventos particulares que afectaran los datos en el año 2016; procurando condiciones de normalidad en los datos observados. Estos datos fueron agregados por intervalos de 15 minutos.

Una vez recolectados los datos, fueron tratados por un equipo de analistas es el encargado de procesarlos, tabularlos y digitalizarlos. El conteo de vehículos se realizó a través de la observación de las horas de grabación por las cámaras instaladas, realizando una clasificación taxis, motos, buses, camionetas y carros particulares; lo anterior puesto que cada grupo tiene sus características como el uso, dimensiones, etc.

El grupo de investigación Sirius con el apoyo del Centro de Innovación y Desarrollo Tecnológico, obtuvieron la información anteriormente descrita en el marco del desarrollo de un modelo de simulación en tiempo real para la zona de estudio, entre los años 2015 y 2016. Esta investigación tuvo la posibilidad de acceder y utilizar la mencionada información, gracias a la participación del autor en la línea de modelado y simulación del grupo de investigación y las gestiones internas realizadas.

4.3. Modelo de microsimulación.

4.3.1. Modelamiento de la red

En esta etapa, se implementa en Transmodeler la oferta; es decir las características propias de la infraestructura vial en la zona de estudio teniendo en cuenta la cantidad de carriles y las dimensiones de las vías. Para las carreras 13 y 14, se modelaron con tres carriles cada una, siendo uno de ellos de uso exclusivo del sistema de transporte masivo de Pereira, Megabus. La calle 21, también cuenta con tres carriles; no obstante, por esta calle no transita el sistema de transporte masivo de la ciudad. Las demás calles implementadas en el modelo son de dos carriles. Por su parte, las dimensiones de las calles y carreras modeladas fueron importadas de manera directa desde el proyecto colaborativo para crear mapas libres y editables conocido como OpenStreetMap a Transmodeler.

4.3.2. Sentidos y giros permitidos.

En esta etapa, se incorporan en la red los sentidos de las vías, giros permitidos y velocidades máximas para los vehículos, las posibilidades en circulación y zonas restringidas. Para la carrera 13, el sentido es de oriente a occidente y este es invertido para la carrera 14, frente las calles con sentido de sur a norte se tienen las calles 19 y 21; mientras que las calles 18, 20 y 22 en sentido opuesto. Se restringe el paso de vehículos particulares o públicos en el carril exclusivo del Megabus. Se impiden giros a la izquierda en la carrera 14, al igual que en la 13. Finalmente, las velocidades máximas están dadas por los límites de velocidad reglamentados por el Instituto de Tránsito y Transporte de Pereira, de máximo 60 km/h en corredores urbanos no residenciales.

4.3.3. Semáforos y sensores.

En la Tabla 3 muestra las intersecciones semaforizadas modeladas en Transmodeler. Donde se expone la cantidad de semáforos que existen en las intersecciones entre carreras y calles para la zona de estudio. En cada uno de estos semáforos se obtuvo el tiempo de fase de las luces verde, naranja y roja.

Tabla 3. Intersecciones semaforizadas.

Carrera	Calle	Semáforos
14	21	2
14	19	2
14	18	2
13	18	2
13	19	1
13	21	2

Fuente: Elaboración propia.

En campo, se cuenta con sensores ubicados como se muestran en la Figura 11, por tanto, en el modelo se localizan sensores en las mismas coordenadas para replicar el conteo de vehículos, en este caso simulados, que pasan por estos puntos. Lo anterior, es importante para la función objetivo a minimizar.

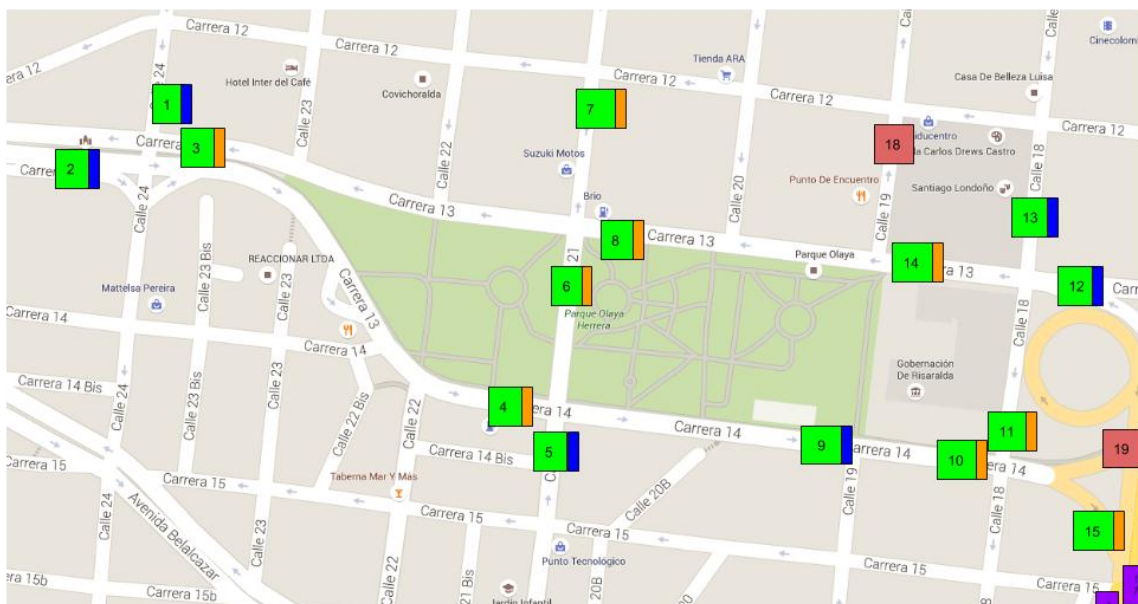


Figura 11. Posición de los sensores. Fuente: Grupo de investigación Sirius.

En la Figura 11 se muestra la disposición de los sensores en campo. Estas son cámaras que se utilizan para grabar la ruta seguida por un vehículo permitiendo con ello identificar el lugar desde donde partió y cual fue el lugar de destino. Los sensores 18 y 19 que aparecen en rojo no se tuvieron en cuenta a pesar de su gran relevancia en la red debido a condiciones externas.

4.3.4. Matriz Origen – Destino, OD.

Con los datos procesados en campo y expuestos en el numeral 4.2 del presente capítulo, se carga los datos tabulados a la red modelada en Transmodeler, para alimentar el modelo en cuánto a la cantidad de vehículos desde diferentes pares orígenes destino, como se aprecia en la Figura 12.

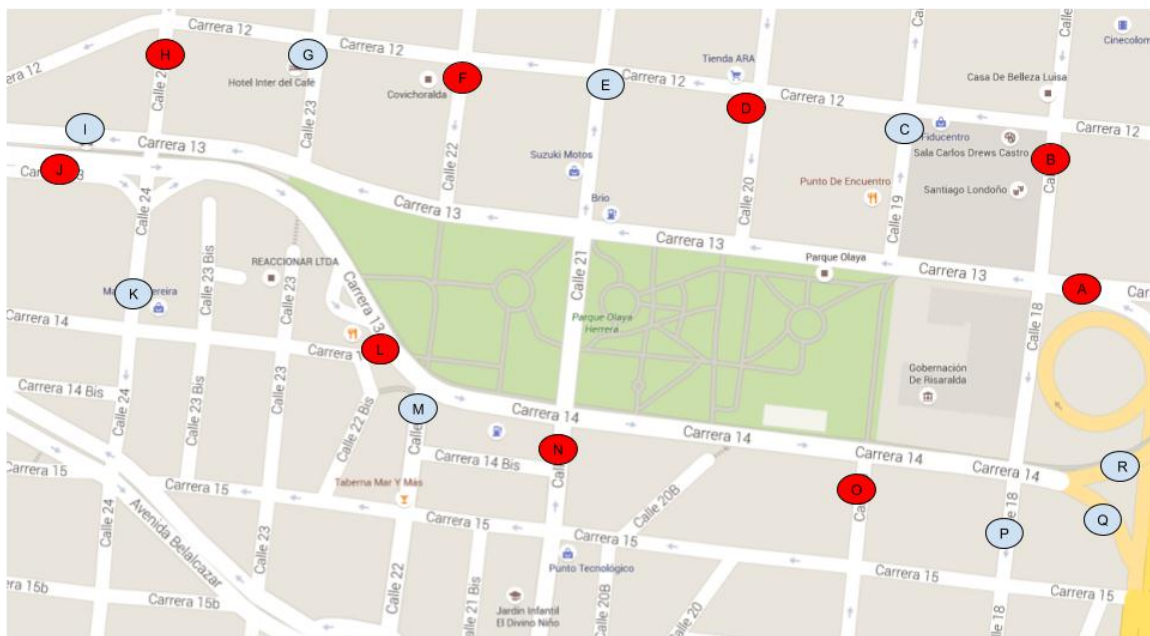


Figura 12. Orígenes – Destinos. Fuente: Grupo de investigación Sirius.

La Figura 12 muestra los puntos utilizados, en color rojo los orígenes mientras que en azul los destinos, para estimar la matriz. Se debe tener en cuenta que no se pueden violar

los sentidos de las vías y se debe seguir una ruta natural. En la Tabla 4 el número uno indica si hay flujo del origen al destino y cero de lo contrario.

Tabla 4. Orígenes – Destinos.

	Destinos								
	C	E	G	I	K	M	P	Q	R
A	1	1	1	1	0	0	1	1	1
B	1	1	1	1	0	0	1	1	1
D	0	1	1	1	0	0	0	0	0
F	0	0	1	1	0	0	0	0	0
H	0	0	0	1	0	0	0	0	0
J	0	0	0	0	1	1	1	1	1
L	0	0	0	0	0	1	1	1	1
N	0	0	0	0	0	0	1	1	1
O	0	0	0	0	0	0	1	1	1

Fuente: Grupo de investigación Sirius.

En la Figura 13 se muestra el resultado final del modelo de microsimulación implementado en Transmodeler para la zona de estudio.

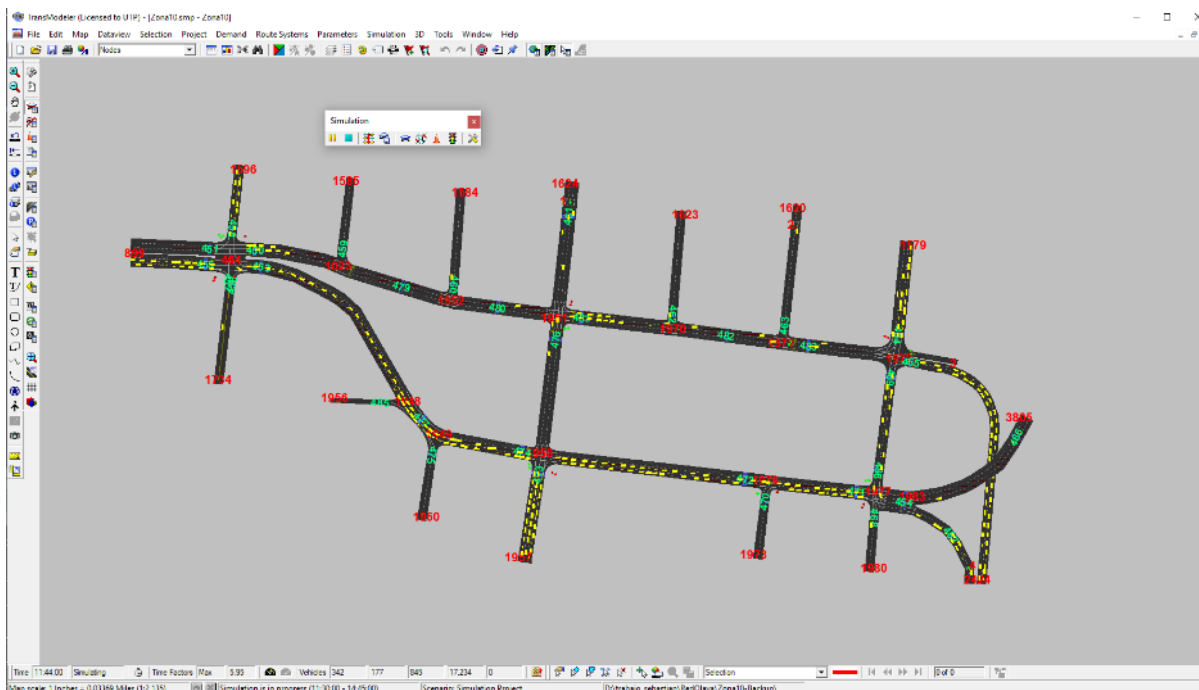


Figura 13. Modelo de microsimulación implementado. Fuente: Grupo de investigación Sirius.

4.4. Software, modelos y parámetros de calibración.

Para calibrar el modelo microsimulación, se precisa describir los elementos tenidos en cuenta para llevar a cabo esta tarea como son: los parámetros seleccionados y los modelos a los que pertenecen, el software de calibración Gauss, la función objetivo-construida, los algoritmos implementados, los lenguajes de programación utilizados y las medidas de rendimiento o MoPs, por sus siglas en inglés de Measure of Performance.

Como se ha expuesto en este trabajo los microsimuladores cuentan con una gran cantidad de parámetros que deben ser ajustados a las condiciones particulares de estudio; Transmodeler no es ajeno a esta situación y cuenta con aproximadamente 2000 parámetros diferentes, lo que se torna en una tarea combinatorial al momento de escoger los mejores valores que ajusten la realidad local; por lo que identificar cuáles son los parámetros que más impactan el comportamiento del modelo son una tarea importante de realizar.

En (Dowling, 2004) se propone que los parámetros a calibrar deben ser aquellos que afecten directamente la capacidad de la red como aquellos que intervienen directamente en el flujo vehicular. Por otro lado, en el trabajo de grado desarrollado por (Jiménez, 2016) se probó como los parámetros Alpha y Gamma del modelo *car following* de Transmodeler afectan directamente los flujos de los vehículos. Estos parámetros pertenecen a modelos que buscan representar las características de conducción que se pueden observar en la realidad, conocidos en la literatura como *Driver behaviour*.

4.4.1. Herramienta computacional: Gauss.

El presente trabajo requiere hacer implementaciones de algoritmos como: genético continuo, enjambre de partículas y SPSA en Transmodeler, con el objetivo de aumentar las técnicas disponibles para hacer calibración. Por lo anterior, se requiere hacer uso de Gauss (Jiménez, 2016) software desarrollado por el grupo de investigación Sirius con el apoyo financiero del Centro de Innovación y Desarrollo Tecnológico CIDT. En la Figura 14 se muestra la arquitectura del software de calibración.

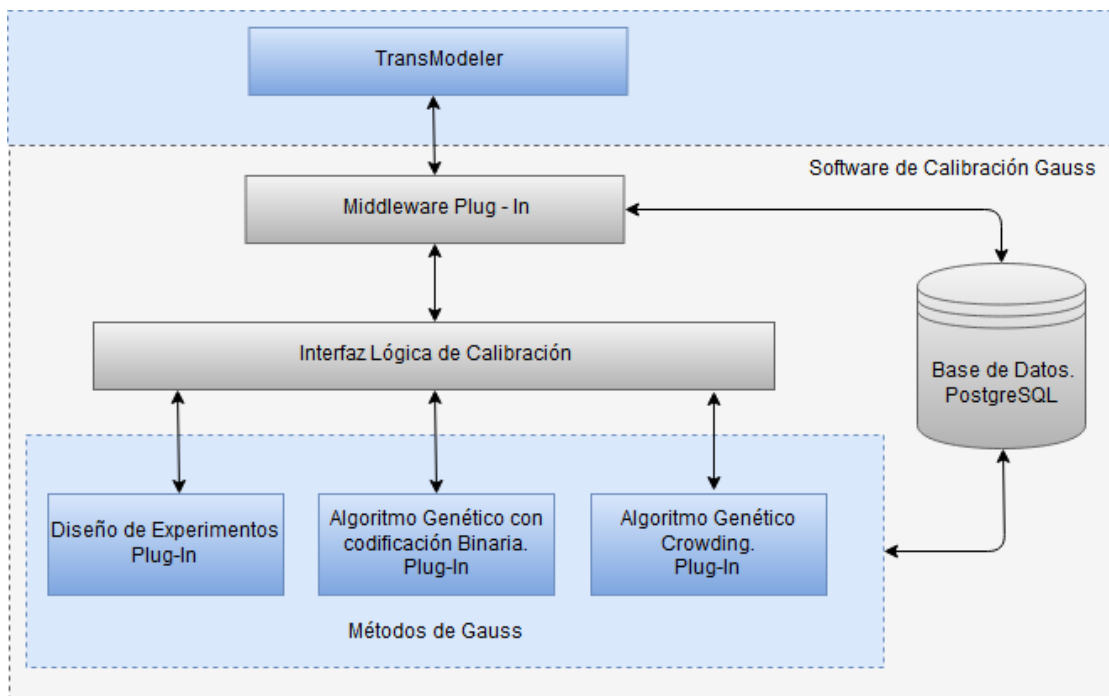


Figura 14. Arquitectura de Gauss. Fuente: (Jiménez, 2016).

Gauss permite controlar de manera automática ciertos eventos de Transmodeler. Por ejemplo, el inicio y fin de la simulación, los parámetros a simular, almacenamiento de salidas en base de datos, cantidad de réplicas, conexiones a otros lenguajes de programación y otras. Todos estos manipulados por medio de líneas de códigos escritas en un lenguaje de programación orientado a objetos conocido como *C Sharp* (C#). Para esto, Gauss utiliza tres módulos: Middleware Plug In, métodos de Gauss y la base de datos.

El Middleware Plug In es el centro del software ya que comunica a los métodos de Gauss y Transmodeler a través de una interfaz lógica de calibración. Los métodos actúan modularmente de tal manera que se puede implementar una amplia gama de algoritmos de manera independiente sin modificar el Middleware. La base de datos es un módulo que permite guardar la información proveniente de Transmodeler que el programador considere relevante para el algoritmo.

El software en la actualidad cuenta con tres métodos: un diseño de experimentos que permite realizar un análisis de sensibilidad de los parámetros que van a ser sujetos al proceso de calibración; dos algoritmos genéticos de estado estacionario para realizar el proceso de calibración, ambos con codificación binaria, uno elitista y el otro *Crowding*. Este último permite mantener la población lo más diversa posible (Mengshoel, 2014).

Este trabajo, implementa en Gauss un algoritmo genético con codificación real, también un SPSA y un enjambre de partículas. La lógica para llevar a cabo un nuevo método en Gauss se implementa a través de la interfaz lógica de calibración, allí se programan todas las rutinas del nuevo método y se definen claramente los pasos ilustrados en la Figura 15.

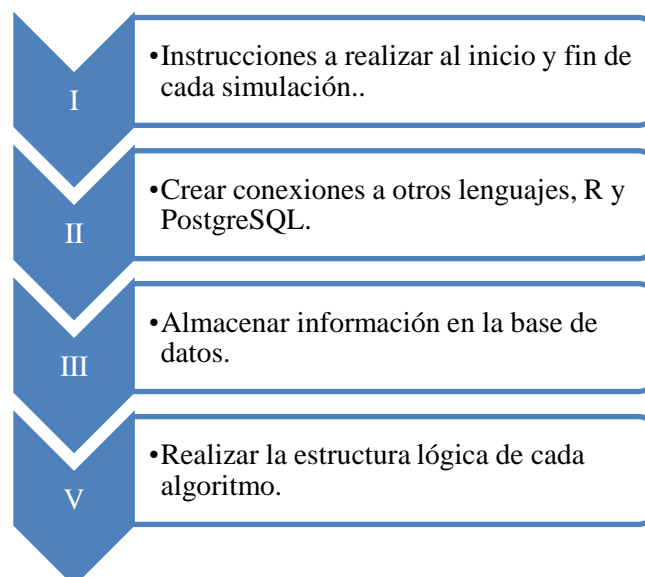


Figura 15. Pasos para implementar un método en Gauss. Fuente: Elaboración propia

4.4.2. Modelo: Car Following

Es un submodelo en Transmodeler que permite representar la forma de acelerar o desacelerar de un vehículo en función de la distancia y la velocidad del vehículo de adelante. La aceleración o desaceleración es determinada por la ecuación (4.4.1) de acuerdo con (Caliper, 2011).

$$A_i^\pm[t + \Delta t] = \alpha^\pm \frac{V_i^{\beta^\pm}[t]}{D_{i,i-1}^{\gamma^\pm}} (V_{i-1}[t] - V_i[t]) + \varepsilon_i^{CF} \quad (4.4.1)$$

A_i^\pm representa la aceleración del vehículo i , V_{i-1} velocidad del vehículo de adelante, V_i es la velocidad del vehículo que realiza el seguimiento $D_{i,i-1}^{\gamma^\pm}$ representa la distancia entre los dos vehículos, los parámetros del modelo se representan por α^\pm , β^\pm , γ^\pm . El error del modelo esta dado por el término ε_i^{CF} . Finalmente, el superíndice \pm indica si se está calculando una aceleración, +, o una desaceleración, -. En la Figura 16 se puede apreciar los vehículos y la distancia entre ellos.

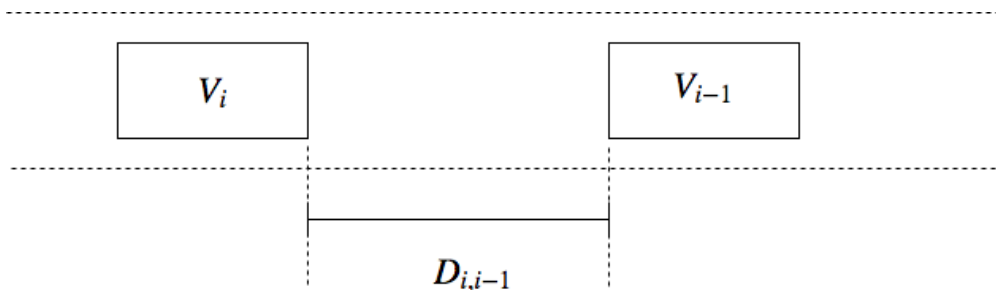


Figura 16. Car Following. Adaptado de: (Caliper, 2011).

En la ecuación (4.4.1) se puede notar que sí la distancia entre a los vehículos es poca el cociente de la expresión se hace grande, lo que implica aplicar una desaceleración para evitar la colisión. Si la distancia por el contrario es grande el cociente es pequeño y habrá de sopesarse con el parámetro Alpha y las diferencias de velocidades para aplicar una aceleración. Como se mencionó anteriormente, en (Jiménez, 2016) se evidencia que los parámetros β y γ son los que más influyen en los flujos y por tanto serán tenidos en cuenta en la presente investigación para realizar la calibración.

4.4.3. Distancias de las Paradas.

Este es otro de los parámetros que influyen de manera importante en la capacidad y comportamiento de los vehículos. Modela la distancia que dejan los autos al de adelante, posibilitando reflejar en el modelo que algunos automóviles se acercan demasiado, mientras otros preferirán dejar más espacio (Caliper, 2011). Este comportamiento, también se ve afectado sí al frente hay un auto pesado, como una tractomula, o uno liviano como una moto.

Este comportamiento es modelado por la ecuación (4.4.2).

$$G_{ik} = \max\{\mu_k + r_i\sigma_k, G_{min}\} \quad G_i > 0 \quad (4.4.2)$$

El índice k indica el tipo de vehículo al frente h si es pesado y n de lo contrario. Por lo tanto G_{ik} es la distancia que deja el vehículo i al k . μ_k es la distancia media que se deja al vehículo de tipo k . r_i es un número aleatorio tomado de una distribución normal estándar. σ_k desviación estándar de la distancia al auto de adelante del tipo k . G_{min} es la mínima distancia que se permite al auto de adelante.

El parámetro que se decidió modificar de este modelo fue μ_k para vehículos no pesados al frente, debido que la flota de vehículos de la zona de estudio es

mayoritariamente liviana. Así mismo, este parámetro se escoge puesto que como promedio es un valor de posición por lo que da información de hacia dónde tiende a ubicarse estas distancias.

4.4.4. Headway.

Existen tres modelos de aceleración: de emergencia, *free flow* y *car following*. El modelo *car following* fue expuesto en el numeral 4.4.2.1. El modelo de emergencia aplica una desaceleración con el fin de evitar una colisión y el modelo *free flow* caracteriza la velocidad a la que desea llegar un vehículo cuando no observa obstáculo adelante y por ende puede aplicar una aceleración constante hasta alcanzar una velocidad deseada.

En la Figura 17 se aprecia el *Headway* que es el tiempo que hay entre un vehículo i e $i - 1$. Establece un límite superior y uno inferior de tiempo para cada modelo, de tal forma que se pueda identificar en cuál de éstos tres se encuentra el vehículo.

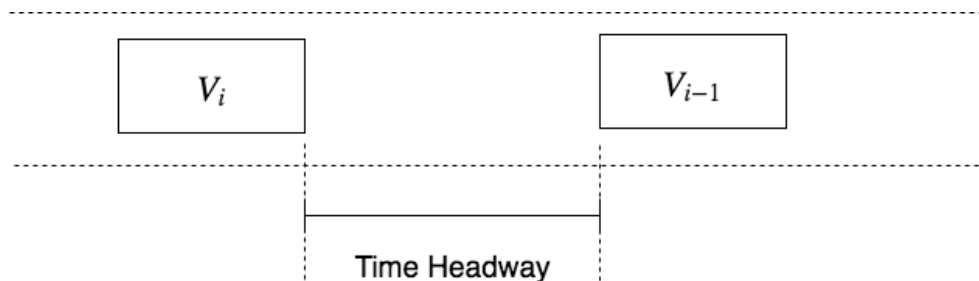


Figura 17. *Headway*. Adaptado de: (Caliper, 2011).

En la Figura 18 se aprecia que si el tiempo *Headway* es menor a un límite inferior el vehículo se encuentra en el modelo de emergencia, si es mayor al límite superior se activa el free Flow, por el contrario, se encuentra en el modo *car following*.

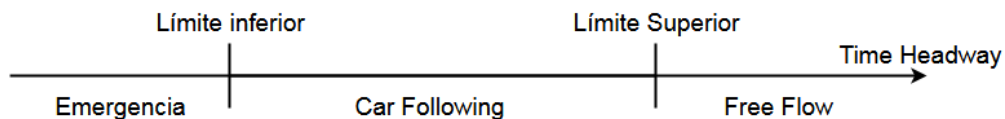


Figura 18. Límites de tiempo Headway. Adaptado de: (Caliper, 2011)

Se selecciona para el presente trabajo, modificar el parámetro de límite inferior para el *Headway*, en concordancia con las condiciones de la zona de estudio. El límite inferior establece un comportamiento conservador que afecta la cantidad de vehículos. Por las razones antes expuestas y debido a que la zona de estudio difícilmente se encuentre en *free flow* por la longitud de la red, el límite inferior es el que puede generar mayor generar mayor impacto en el estudio.

En la Tabla 5 se resumen los parámetros seleccionados para calibrar en el presente trabajo, así mismo se identifican con el modelo al que pertenecen y los valores que por defecto trae Transmodeler. También se exponen los límites máximo y mínimo que pueden tomar los parámetros en esta investigación; estos límites se escogen de acuerdo con el juicio de expertos en el desarrollo de estos modelos en Transmodeler.

Tabla 5. Parámetros por calibrar.

N°	Parámetro	Modelo	Valor Defecto	Límite Superior	Límite Inferior
1	β aceleración	Car Following	-1.67	4	-5
2	γ aceleración	Car Following	-0.89	4	-5
3	Límite Inferior	HeadWay	0.3 s	12 s	0.1 s
4	μ_k vehículos pesados enfrente	no Stop Gaps	0.60 m	5 m	0.01 m

Fuente. Transmodeler y propia.

4.5. Variables de campo para calcular una medida de discrepancia (*MoP Measures of performance*).

Para medir la discrepancia entre la realidad y los resultados del simulador se requiere construir una función objetivo en función de los MoP reales y simulados. Los simulados se obtienen directamente del simulador más los reales hay tomarlos en campo.

Por lo tanto, para llevar a cabo un proceso de calibración de manera exitosa es elegir estas variables. Para ello se debe tener en cuenta el estudio deseado; por ejemplo, las medidas serán unas sí se desea evaluar el impacto de una señalización antes de ingresar en una glorieta u otras, sí lo que se busca es cambiar el sentido de una vía. En general, algunas de las variables más observadas en campo hacen referencia al tiempo de viaje, velocidades, distancias entre vehículos, longitudes de colas, etc. Obtener esta información por lo general es costoso ya que se requiere de tecnologías y personas que usualmente no se tiene a la mano.

Como se ha expuesto a lo largo del trabajo, el grupo de investigación Sirius con el apoyo del CIDT pudo recolectar información de campo, entre ésta la concerniente a los flujos vehiculares necesarios para adelantar el proceso de calibración en este proyecto.

4.6. Función Objetivo.

En la calibración a realizar no es posible contar un modelo analítico que relacione todos los parámetros del simulador y las salidas de este. El proceso de calibración busca minimizar la diferencia entre las variables obtenidas en campo y las del simulador, para el caso los flujos vehiculares. Existen diferentes medidas para cuantificar la discrepancia, en la literatura se denominan como *GoF*, goodness of fit, como bien se muestra en (B Ciuffo, 2012b). Esta función y el algoritmo implementado influyen en las respuestas que se puede obtener del proceso de calibración. En la presente investigación se consideró la minimización de la diferencia de cuadrados media; esta medida tiene la ventaja de penalizar diferencias grandes, gracias a la suma de cuadrados como se muestra en la ecuación (4.4.3). Cabe anotar que la diferencia de cuadrados media es la función objetivo más utilizada en la literatura.

$$GoF = \frac{\sum_{i=1}^{ni} \sum_{j=1}^{ns} (MoPReal_{i,j} - MoPSimu_{i,j})^2}{ni * ns} \quad (4.4.3)$$

$MoPReal_{i,j}$ es el MoP real tomado en el intervalo i sensor j . Estos son los flujos tomados en campo. $MoPSimu_{i,j}$ es el mismo MoP pero tomado esta vez de modelo de simulación. ni representa la cantidad de intervalos de tiempo tenidos en cuenta para tomar la información. ns es la cantidad de sensores que se dispuso en campo.

4.7. Descripción detallada de los algoritmos implementados.

Los algoritmos tenidos en cuenta son: un algoritmo genético de estado estacionario con codificación real, este debido a que Gauss cuenta con dos AG pero con codificación binaria. Un SPSA el cual es atractivo no solo porque es ampliamente utilizado en la literatura, sino también porque permite estimar el gradiente. Por último, un enjambre de partículas que ha sido poco abordado en la literatura y cuyas propiedades hacen de este un algoritmo prometedor.

4.7.1. Algoritmo genético.

El algoritmo genético implementado es uno de estado estacionario con codificación real. Este puede ser observado en el diagrama de flujo de la Figura 19. Allí se aprecia como para implementar un AG en el proceso de calibración se requiere una sincronización entre el algoritmo y Transmodeler. Se crea un individuo aleatorio en el espacio de soluciones, luego esta información es ingresada de manera automática por medio de Gauss a Transmodeler para realizar la simulación (recuadro verde de la misma figura), una vez finalizada la información del individuo simulado se guarda en la base de datos. Este proceso se repite hasta que en la base de datos hay tantos individuos como en el tamaño de la población.

Luego de que la población inicial es simulada se realizan los pasos del AG, (recuadro amarillo de Figura 19), la población inicial es consultada en la base de datos para realizar dos torneos y así seleccionar dos padres; estos son cruzados de acuerdo con la tasa y mediante el operador de cruzamiento plano. La mutación es uniforme conforme con la tasa establecida.

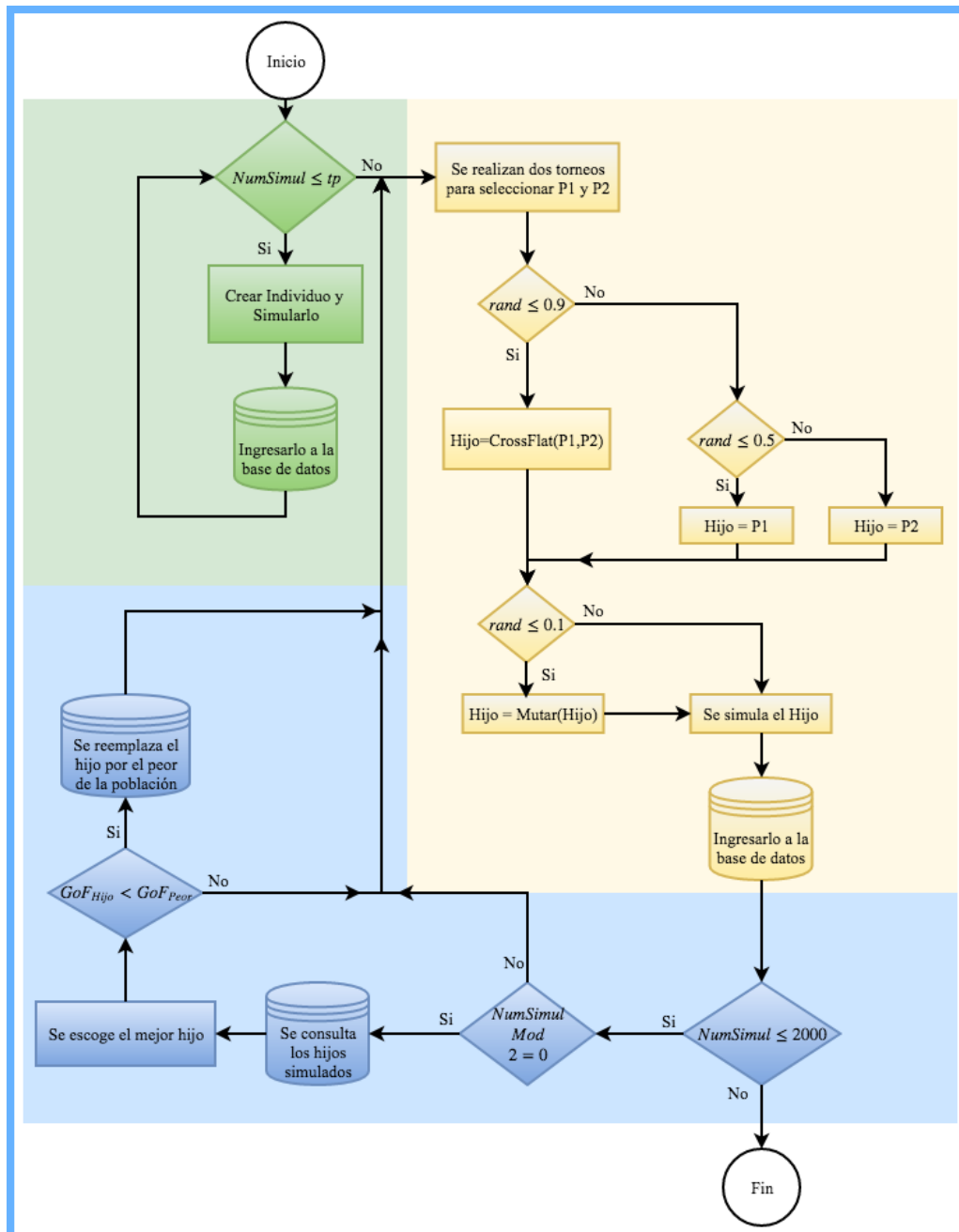


Figura 19. Implementación del AG. Fuente: Elaboración Propia

El nuevo individuo es simulado para luego ser almacenado en la base de datos. Estos pasos se repiten para crear otro individuo o segundo hijo. Para seleccionar el hijo mejor adaptado se consulta la base de datos y se comparan las funciones de adaptación de los dos últimos generados, el ganador se enfrenta con el peor de la población y en caso de ser mejor lo reemplaza. Los anteriores procesos, se repiten hasta alcanzar un criterio de parada, para el caso dos mil simulaciones, recuadros azules.

A continuación, se exponen las consideraciones particulares de los pasos generales para implementar un algoritmo genético ilustradas en la Figura 19.

4.7.1.1. Codificación.

Debido a que la codificación real es más natural para el tipo de parámetros que se están abordando en el problema bajo estudio, esta será la utilizada. Cada individuo tiene cuatro posiciones que representan los parámetros a calibrar seleccionados en el numeral 4.4.2. El primera y segunda posición son para β y γ del modelo *car following* en aceleración; la tercera es el límite inferior para el tiempo *Headway* y la última posición hace referencia a la distancia de parada μ_k que dejan los carros al de adelante cuando están en reposo. A manera de ejemplo se tiene la ecuación (4.4.4) que muestra la codificación del individuo, donde cada de sus componentes representa los valores dados a cada uno de los parámetros.

$$x = [-2.54 \quad -1.3 \quad 1.45 \quad 0.9] \quad (4.4.4)$$

4.7.1.2. Población.

La población es creada uniformemente sobre el espacio de soluciones, respetando los límites máximos y mínimos para cada parámetro dados por la Tabla 5. Esta población es simulada para obtener el *GoF* de cada uno de los individuos.

4.7.1.3. Selección Torneo.

Para el proceso de selección de los dos padres se realizan dos torneos determinísticos con reemplazo utilizando una muestra de tres individuos; es decir, para el primer torneo se seleccionan aleatoriamente tres y de estos se escoge el mejor adaptado. Es con reemplazo debido a que los tres anteriores vuelven a la población para ser candidatos en el segundo torneo.

4.7.1.4. Cruzamiento.

Como se expresó en el marco teórico se va a realizar un cruzamiento plano. Esto es, si se tienen dos padres P_1 y P_2 con valores como los que se muestran en las ecuaciones (4.4.5) y (4.4.6), el hijo va a ser creado como un número aleatorio uniformemente distribuido entre los límites de cada una de las componentes de ambos padres. Es decir, para el ejemplo un número entre -2.54 y -1.54, otro entre -1.3 y 2.5; y así sucesivamente.

$$P_1 = [-2.54 \quad -1.3 \quad 1.45 \quad 0.9] \quad (4.4.5)$$

$$P_2 = [-1.54 \quad 2.5 \quad 3.11 \quad 1.9] \quad (4.4.6)$$

4.7.1.5. Mutación.

Luego de que los padres son cruzados el hijo puede ser mutado, de acuerdo con la tasa de mutación. Para este proceso, se selecciona aleatoriamente una posición del hijo y luego se genera un número aleatorio uniforme entre los límites superiores e inferiores del parámetro asociada a la componente seleccionada.

4.7.1.6. Reemplazo.

En el reemplazo, si el hijo tiene mejor adaptación que el peor individuo de la población entra a tomar su posición.

4.7.1.7. *Criterio de parada.*

El algoritmo se detiene cuando se cumple dos mil simulaciones. Este criterio considera una limitante técnica en Gauss, la cual es que una vez ingresado el número de simulaciones a realizar, el software no para hasta finalizarlas.

4.7.1.8. *Configuración del AG.*

Finalmente, en la Tabla 6 se resumen los valores utilizados para la configuración del algoritmo genético implementado en la presente investigación.

Tabla 6. Configuración implementada en AG.

Configuración	Valor
Tamaño de la Población	20
Tamaño del Torneo	3
Cantidad de Torneos	2
Tasa de cruzamiento	0.9
Tasa de Mutación	0.1

4.7.2. **Enjambre de partículas.**

El enjambre de partículas fue implementado al igual que el algoritmo genético teniendo en cuenta las simulaciones como referencia para ejecutar la lógica de programación. En la Figura 20 se muestra el diagrama de flujo para este algoritmo, lo primero a realizar es simular el enjambre inicial (recuadro azul); de manera que, si el número de simulaciones, el cual es un contador que opera en Gauss, es menor a la cantidad de partículas se crea el enjambre y se ingresa a la base de datos. Cuando el enjambre es creado, se empieza a actualizar las posiciones y velocidades de cada una de las partículas hasta que la cantidad de simulaciones llegue a 2000 (recuadro gris). De esta manera cada vez que una de las partículas se va a actualizar se consulta en la base de datos su mejor posición y la mejor partícula del enjambre.

4.7.2.1. Codificación.

Al igual que el algoritmo genético, para el PSO se realiza una codificación real para cada partícula en la que las coordenadas de cada una de ellas están dadas por un vector de variables reales. Tal como se expuso en la codificación de los algoritmos genéticos, numeral 4.4.5.1.1. del presente documento. Con esta codificación se puede escribir en un programa, para el caso C#, los pasos que se explican en el numeral 3.3 de este documento referentes al marco teórico del algoritmo enjambre de partículas.

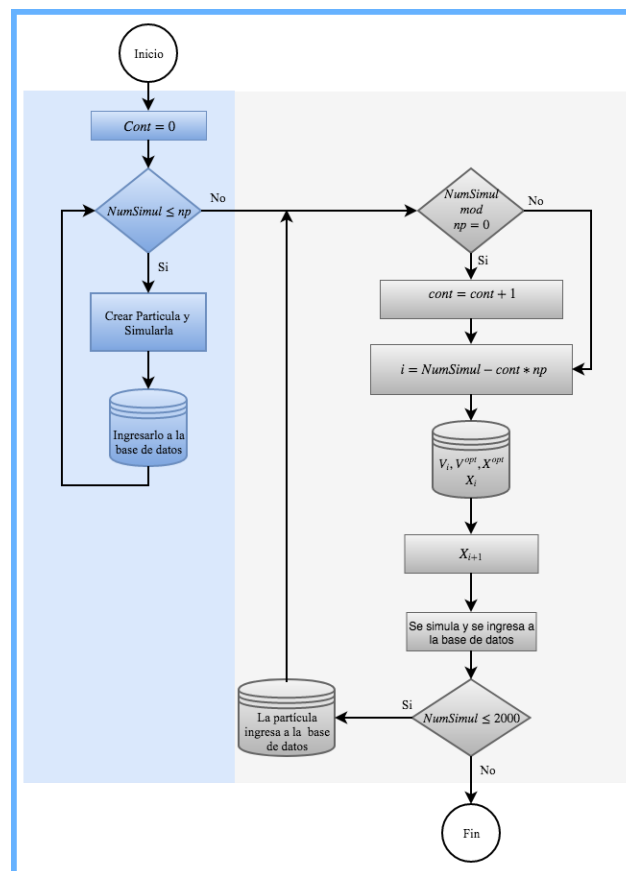


Figura 20. Implementación PSO. Fuente: Elaboración Propia.

4.7.2.2. Penalización de la función objetivo.

En muchas ocasiones, las partículas pueden salir de espacio de soluciones y con ello llevar a un menor valor en la función objetivo; de tal forma que los parámetros son inconsistentes, por ejemplo, números negativos para el tiempo. Por tal motivo se penaliza proporcionalmente a la cantidad de parámetros que violan sus restricciones logrando que la posición no sea atractiva por el enjambre y evitando explorar regiones infactibles.

4.7.2.3. Configuración del PSO.

En la Tabla 7 se resumen los valores utilizados para la configuración del PSO en la presente investigación. Estos fueron obtenidos por medio de pruebas y realizando simulaciones que permitieron establecer un comportamiento adecuado del algoritmo.

Tabla 7. Valores configuración PSO.

Configuración	Valor
Tamaño del Enjambre	20
Máximo Inercia	0.2
Mínimo de Inercia	0.7
Componente Social	1
Componente Cognitivo	1
R	$U(0,1)$

4.7.3. SPSA.

El SPSA es un algoritmo que requiere un punto de inicio. En el presente trabajo, este punto de inicio está dado por los valores de los parámetros que por defecto tiene Transmodeler, mostrados anteriormente en la Tabla 5.

De acuerdo con la ecuación (3.4.1) expuesta en el marco teórico del algoritmo SPSA, se precisas de un punto de inicio $\hat{\theta}_k$ para estimar el valor óptimo θ^* , para el caso de

estudio dicho punto de inicio se presenta en la ecuación (4.4.7). Esta posición es actualizada mediante los pasos dados en el marco teórico.

$$\hat{\theta}_k = [-1.67, -0.89, 0.6, 0.3] \quad (4.4.7)$$

Para este algoritmo, el punto de inicio es perturbado una cantidad C_k en todas las coordenadas tanto positiva como negativamente. Posteriormente se estima el gradiente para restárselo a la posición inicial, cuando se trata de un problema de minimización. Se ingresan las posiciones actualizadas del gradiente en la base de datos. El diagrama de flujo de este método se muestra en la Figura 21.

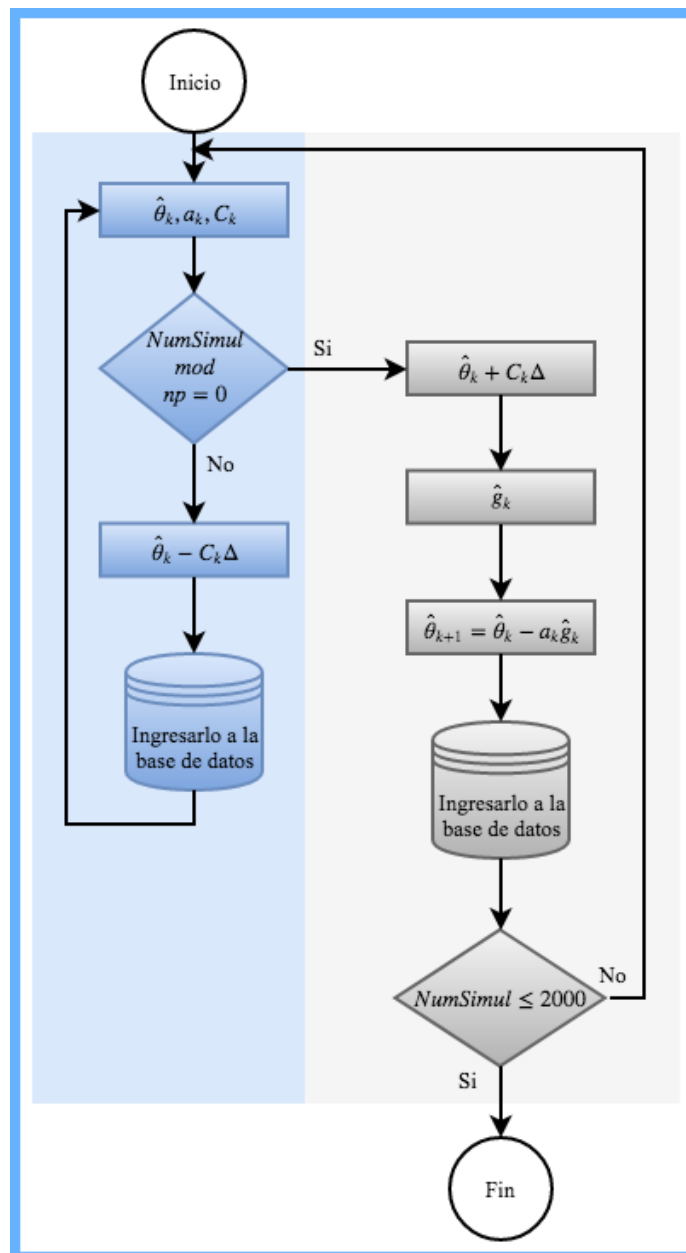


Figura 21. Implementación SPSA. Fuente: Elaboración Propia.

4.7.3.1. Configuración del SPSA:

En la Tabla 8 se resumen los valores utilizados para la configuración del SPSA en la presente investigación. Estos fueron obtenidos por medio de pruebas y realizando simulaciones que permitieron establecer un comportamiento adecuado del algoritmo.

Tabla 8. Valores de la configuración del SPSA.

Configuración	Valor
a	0.16
A	0
c	3
α	0.602
γ	0.101

4.8. Características para tener en cuenta al realizar calibración.

4.8.1. Semilla.

Debido a que Transmodeler es un simulador, se debe garantizar que cada corrida genera una muestra aleatoria. Este software hace el cambio de semilla de forma automática, de tal manera que por cada corrida hay resultados diferentes, aunque la corrida tenga la misma configuración para todos sus parámetros.

4.8.2. Cantidad de réplicas.

En consideración con la naturaleza aleatoria del software, se requieren realizar réplicas de cada configuración de parámetros ya que un mismo conjunto genera valores

diferentes para la variable de interés, flujo vehicular. Por tal motivo se estima el promedio con una cantidad de réplicas de 10, esto genera un error inferior al 1% con una probabilidad de 0.95. Se realiza una estimación de la desviación estándar con algunas simulaciones previas y se hace uso de la fórmula que se aprecia en el marco teórico (3.9.1).

4.8.3. El periodo de calentamiento.

Este periodo en simulación se utiliza para buscar estabilizar el sistema; para el caso bajo estudio se toma de diez minutos, los cuales transcurren antes de iniciar con la toma de datos de la simulación, garantizando que la red este cargada.

5. Resultados

En este capítulo se presentan los resultados de la presente investigación al calibrar un modelo de microsimulación en Transmodeler por medio de las tres estrategias expuestas en el capítulo 4. Cabe resaltar que uno de los puntos fuertes de la presente investigación es que el proceso de calibración se realiza con datos reales, esto es relevante debido a que la mayoría de las investigaciones que se llevan a cabo en el área son realizadas con datos sintéticos dado que los investigadores no tienen acceso a información de campo debido a los recursos de tiempo y dinero que esta actividad requiere. Esta labor fue realizada por el grupo de investigación Sirius con el apoyo del Centro de Innovación y Desarrollo Tecnológico de Pereira.

Los resultados siguientes son producto de realizar 10 procesos de calibración por algoritmo. Cada proceso de calibración consta de 2000 simulaciones; para estimar los flujos y compararlos con los reales cada conjunto de parámetros es simulado 10 veces para luego promediar las salidas del simulador; es decir, se tienen como máximo 200 configuraciones

para los parámetros. El tamaño de muestra fue elegido de 10 debido a que permite estimar la media con un error del 1% y una probabilidad del 0.95. Este proceso es arduo debido a que cada proceso de calibración requiere de alrededor de 60 horas.

El capítulo inicia con un análisis general de los resultados obtenidos en el proceso de calibración por los tres algoritmos y posteriormente se realiza un análisis individual de cada uno de los métodos implementados que describe el comportamiento de las estrategias en la búsqueda de las soluciones. Además, se presenta un gráfico de convergencia que muestra cómo evoluciona la búsqueda a medida que el número de simulaciones aumenta. Por último, se caracteriza si los valores encontrados por el proceso de calibración son mejores que los valores que trae el software por defecto.

Todos los algoritmos fueron implementados en los lenguajes R, C sharp y posgres sql, usando un servidor con procesador Intel Core (TM) i7 X 980, 3.33 GHz, 32 GB de RAM, 2 TB de almacenamiento y sistema operativo Windows 10 64 bits.

5.1. Resultados generales

La Figura 22 muestra los resultados del proceso de calibración para todos los algoritmos implementados. En esta se representa información de 5 variables en un espacio de tres dimensiones. Los ejes x, y, z representan los parámetros β , γ y Headway expuestos en la Tabla 5. El tamaño de la viñeta en la figura se relaciona con valor para el parámetro 4,

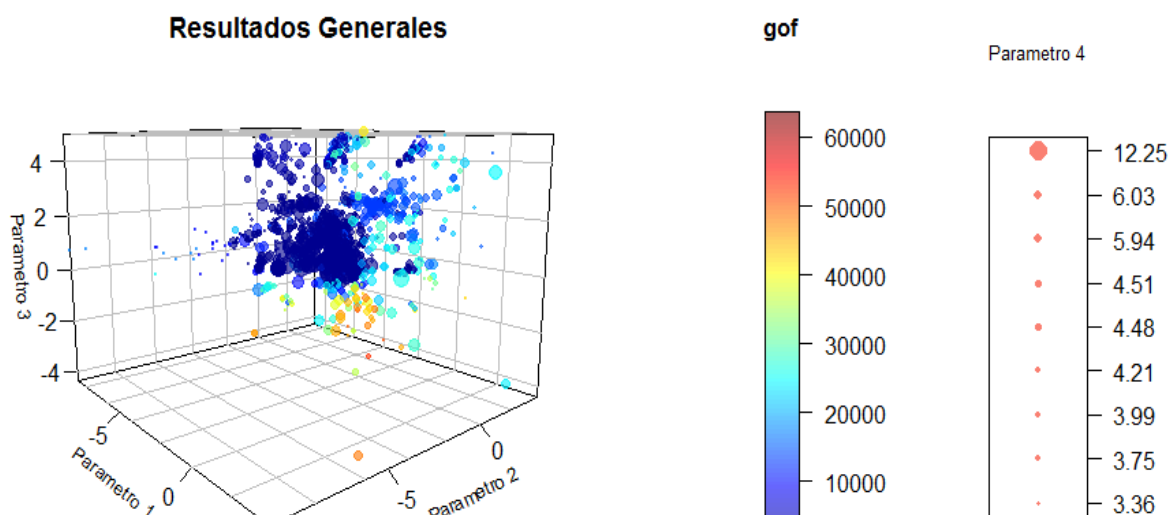
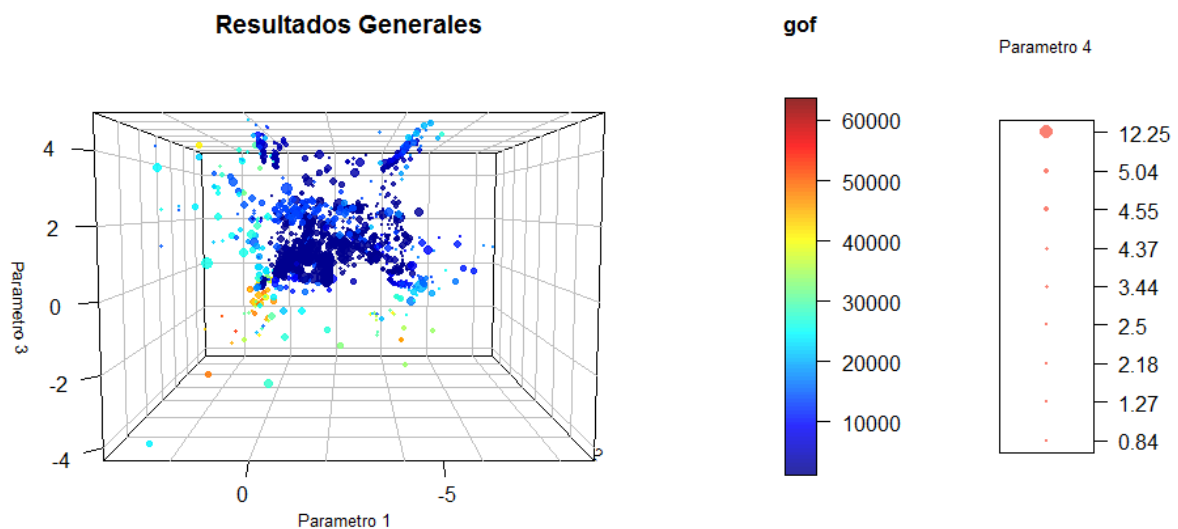


Figura 22. Resultados generales de los algoritmos implementados. Fuente: Elaboración Propia.

es decir la distancia que se deja con respecto al auto de adelante cuando los vehículos se encuentran detenidos, finalmente el color está asociado con el valor de la función objetivo.

Esta figura, permite de manera descriptiva identificar patrones asociados con los valores de los parámetros y la variable de respuesta GoF o función objetivo. En la Figura 22 se puede apreciar como la gran mayoría de las viñetas tiene color azul oscuro, esto se debe a que los algoritmos como el PSO y el AG reemplazan las peores soluciones y por lo tanto se van dirigiendo a lugares donde se presentan buenas respuestas. Se puede notar como para valores negativos del parámetro 3 los colores son claros, entre 30.000 y 50.000, indicando que valores negativos no son atractivos; además de que son pocos los puntos en la región donde el parámetro 3 es menor a cero. Esto es apenas esperado ya que este parámetro está asociado con el tiempo y tiene una penalización si este es negativo.

La Figura 23 muestra la perspectiva del parámetro 1 o beta. Allí se observa que valores positivos para este aumentan la función objetivo. En esta región se tienen diferentes tamaños para el parámetro 4 y diferentes alturas para el parámetro 3, lo que sugiere que el parámetro 1 puede impactar el comportamiento de la red de manera que la discrepancia entre lo simulado y la realidad aumente, provocando que la cantidad de puntos que allí sean pocos, haciendo de esta zona no atractiva para la búsqueda.



En Figura 24 se aprecia que con el parámetro 2 ocurre algo similar. Los valores positivos inferiores a 2 se pueden llegar a buenas soluciones siempre y cuando el parámetro 3 se mantenga positivo, el parámetro 4 sea pequeño y el parámetro 1 sea menor a cero. Aunque también cabe notar que entre más positivo el parámetro 2 la calidad de la solución disminuye.

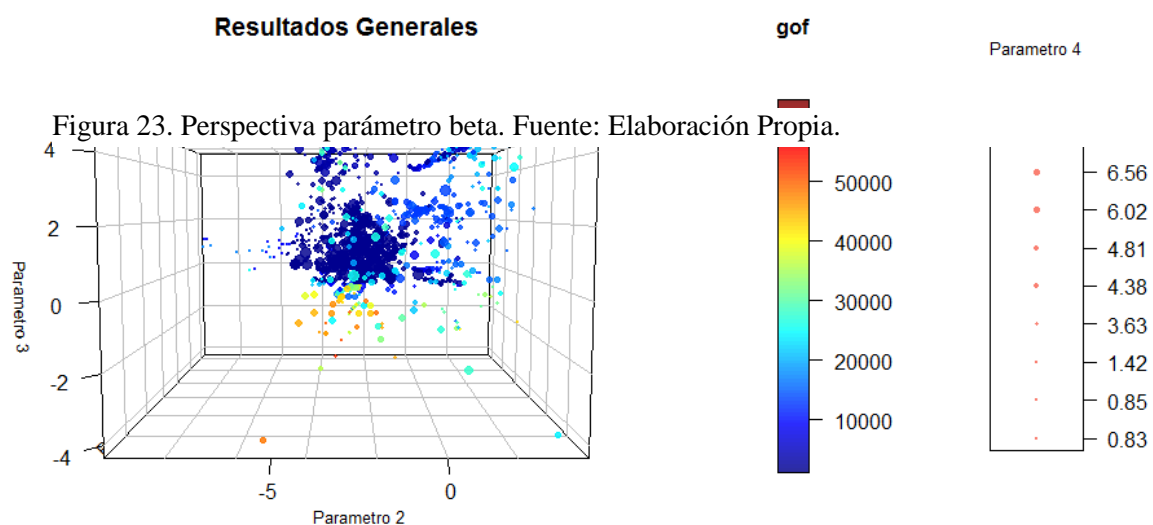


Figura 24. Perspectiva parámetro gamma. Fuente: Elaboración Propia.

Para el parámetro 4 asociada con la distancia al auto de adelante y representado con el tamaño de la viñeta no se nota que un tamaño domine. Se observan tamaños grandes y pequeños para diferentes evaluaciones de la función objetivo lo que puede indicar una interacción con los demás.

Por último, en la Figura 25 se toma una muestra aleatoria de 400 puntos de los resultados con el fin de evitar la superposición en la nube de puntos y constatar las descripciones anteriormente expuestas.

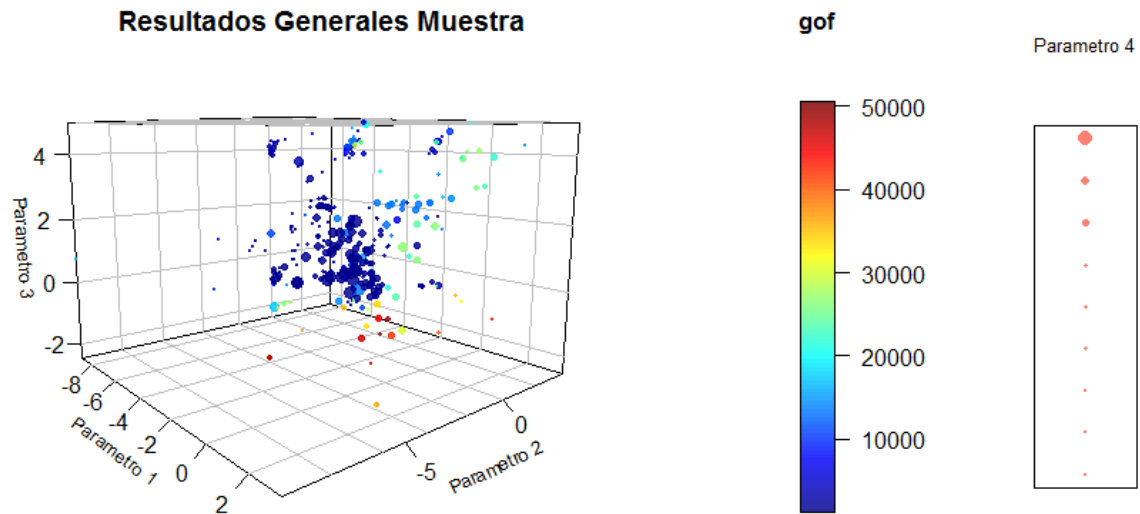


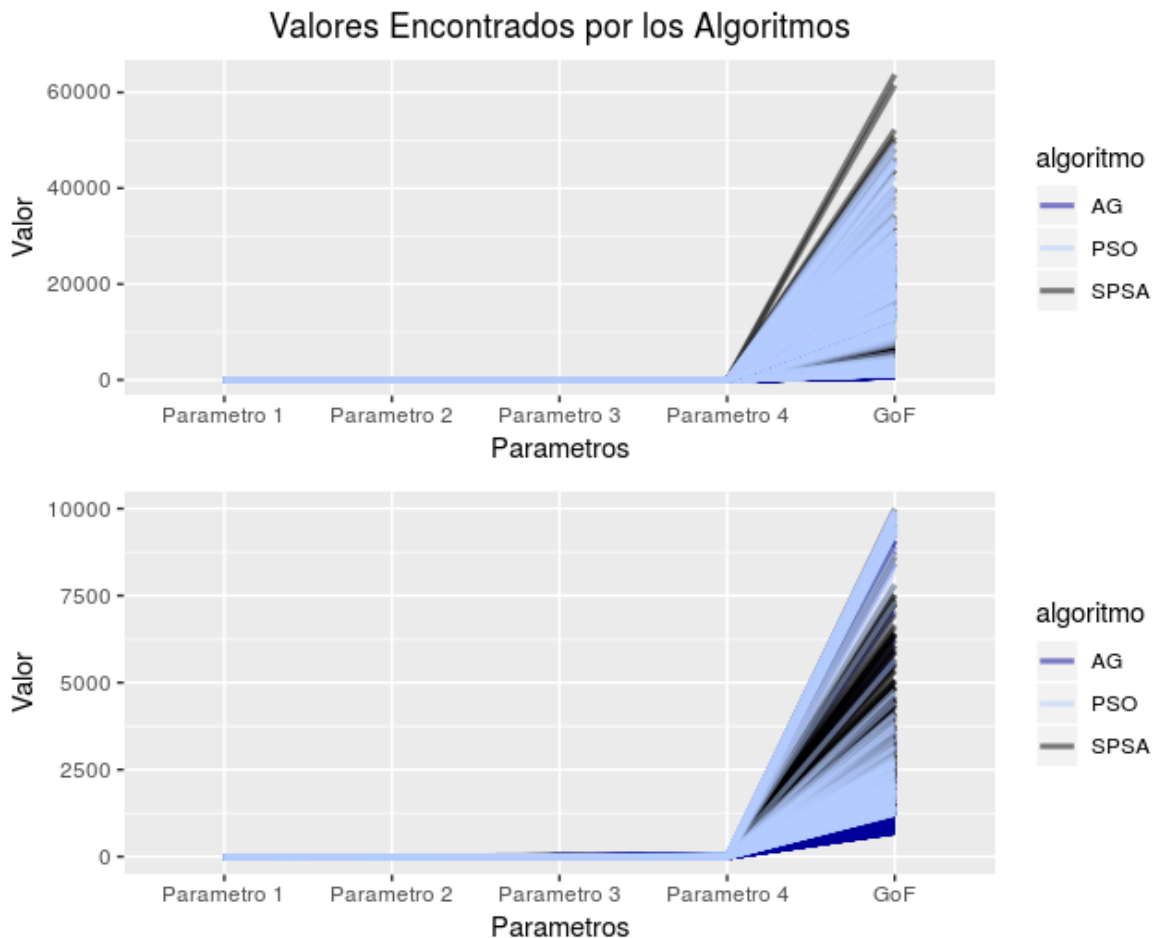
Figura 25. Muestra de los resultados. Fuente: Elaboración Propia.

Las figuras anteriores muestran claramente que el espacio de soluciones cuenta con diferentes regiones, algunas grandes y otras pequeñas, que presentan buenos resultados para la función objetivo, algo que se presumía a partir del estado del arte y es que el problema de calibración tiene múltiples soluciones con un núcleo fuertemente definido.

5.2. Búsqueda realizada por los algoritmos

En el numeral 5.1 se realizó una descripción de los resultados obtenidos al realizar el proceso de calibración para los algoritmos abordados en el presente trabajo. En este apartado se busca caracterizar el comportamiento que siguieron los métodos al momento de realizar el proceso de optimización. Por lo anterior, se exponen diagramas de coordenadas paralelas tal y como se aprecia en la Figura 26. En esta se pueden notar los valores obtenidos en el GoF por cada uno de los métodos implementados. Debido a que los valores del GoF son mayores comparados con los valores de los parámetros no se logra identificar un patrón; aunque si logra notarse que el AG logra obtener los valores más bajos para el GoF. En el panel superior se nota como los peores valores para la función objetivo fueron

obtenidos con el SPSA. En el panel inferior solo se tiene en cuenta los valores para el ajuste inferiores a 10000, el enjambre obtiene resultados en un rango amplio, aunque no logra igualar los resultados del AG.



Ahora bien, en la Figura 26 se nota claramente que el AG fue el que mejores resultados encontró para el GoF lo que lleva a preguntar: ¿Por qué? Para tratar de resolver la pregunta anterior se busca identificar si los algoritmos siguieron algún patrón en su búsqueda y para ello se realiza una transformación lineal a los datos para que sus valores se

Figura 26. Valores encontrados por los algoritmos. Fuente: Elaboración Propia.

encuentren entre 0 y 1 de manera que cada una de líneas represente una solución. Esto permite identificar como se comportaron los algoritmos al realizar la optimización.

El comportamiento de cada algoritmo se puede apreciar mejor, sí se discrimina cada uno de ellos, por esto es de esperar que los comportamientos sean diferentes debido a que cada uno presenta sus propios argumentos y enfoques metodológicos.

5.2.1. Resultados para el SPSA

En la Figura 27 se muestra el grafico de coordenadas paralelas para el SPSA donde se aprecia que este algoritmo es muy consistente en su búsqueda. Exhibiendo un patrón en el comportamiento muy similar para las simulaciones donde se llevó a cabo este método. También se nota como diferentes combinaciones de parámetros dan valores para la función objetivo que se pueden considerar bajos.

Esta consistencia en la búsqueda puede deberse a que el método requiere un punto de inicio, para el caso del presente estudio son los valores por defecto que tiene el microsimulador, desde donde buscar la dirección de descenso. Estas direcciones están dadas por la perturbación aleatoria que siguen una distribución de Bernoulli. Esto puede ser un desacierto ya que no se realiza una exploración del espacio de soluciones lo suficientemente amplia y que puede poner este algoritmo es desventaja con sus competidores. Aunque es algo de esperar ya que esta característica la padecen la gran mayoría de métodos basados en gradiente.

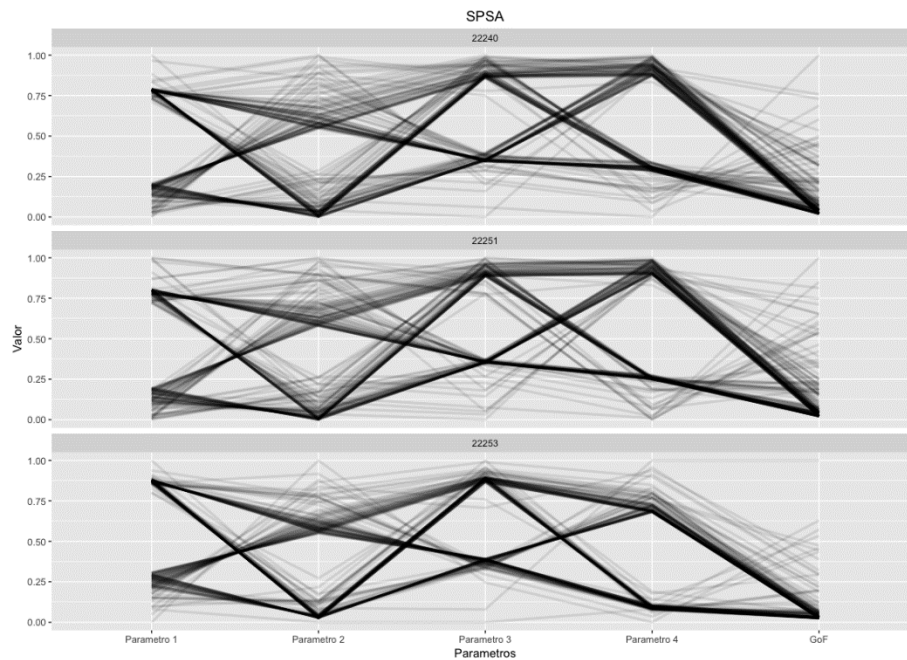


Figura 27. Coordenadas paralelas SPSA. Fuente: Elaboración Propia.

5.2.2. Resultados para el Algoritmo genético

En la **¡Error! No se encuentra el origen de la referencia.** se puede apreciar los resultados del algoritmo genético, en este se ve como los valores para las diferentes simulaciones tienden a concentrarse alrededor de aquellos que ofrecen buenas respuestas para la función objetivo. En la figura del centro se puede ver como hay dos valores para el parámetro cuatro que dan buenos resultados en la función objetivo, este resultado es interesante ya que se puede argüir que el algoritmo puede mantener diversos óptimos locales. Se nota consistencia en la búsqueda argumentado por el agrupamiento de las líneas que tienden a cerrarse en ciertos valores lo que indica convergencia a óptimos locales.

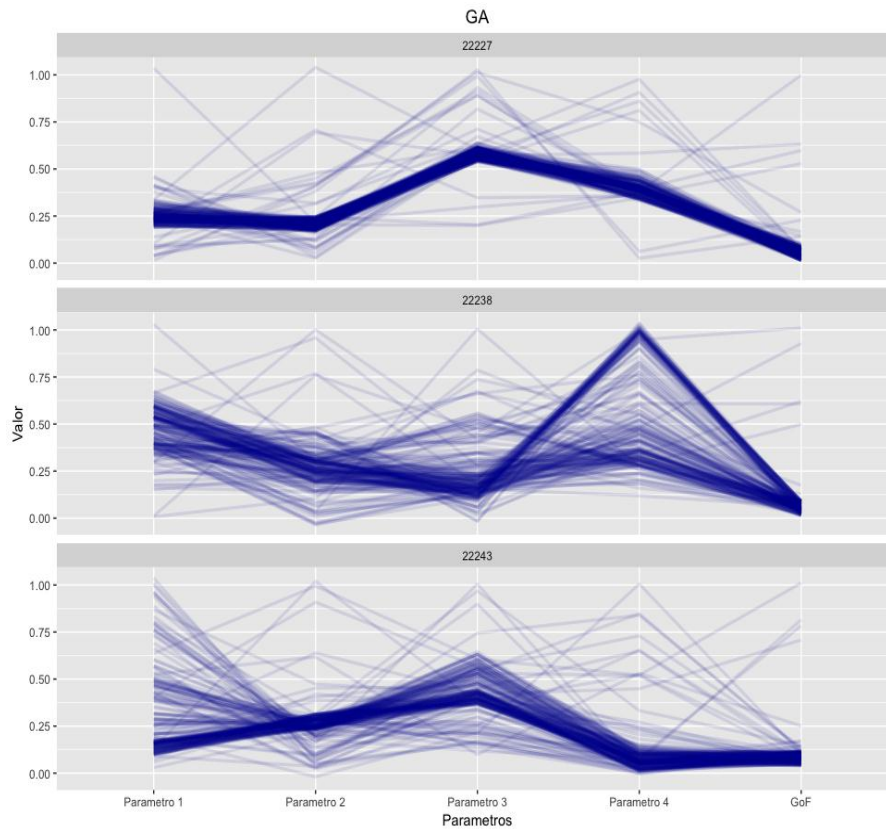


Figura 28. Coordenadas paralelas AG. Fuente: Elaboración Propia.

5.2.3. Resultados para el PSO

En la Figura 29 se muestran los resultados del comportamiento del enjambre de partículas. Esta, comparada con las dos anteriores, presenta una mayor dispersión en sus líneas en sus tres paneles indicando con ello una mejor exploración del espacio de soluciones, además puede notarse una tendencia en los valores centrales que llevan a buenos resultados de la función objetivo. Este comportamiento es esperado ya que como las partículas tienen información de la mejor posición encontrada por el enjambre y la posición propia permite mayor diversidad sin perder de vista la mejor solución encontrada hasta el momento.

No obstante, la diversidad no necesariamente es una ventaja en el proceso de calibración, ya que las simulaciones consumen tiempo, por lo que puede preferirse una técnica que explore mejor un óptimo local para lograr un mejor ajuste.

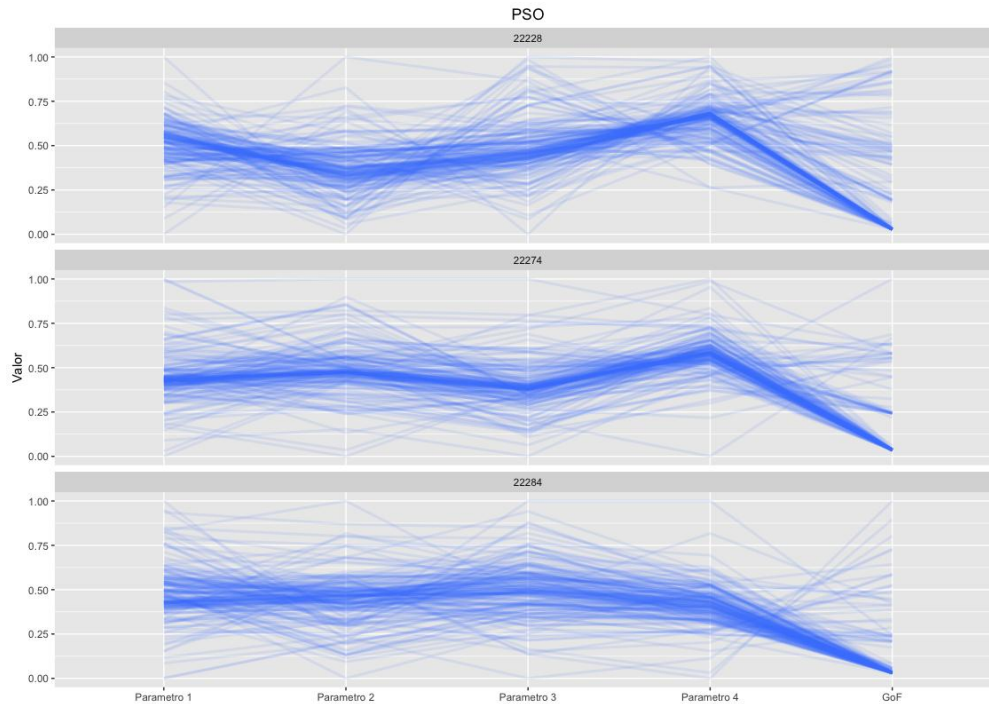


Figura 29. Coordenadas paralelas PSO. Fuente: Elaboración Propia.

De la Figura 28 y la Figura 29 se nota que el AG tiene a concentrarse alrededor de los valores que originan un buen GoF con mayor consistencia que el PSO esto le permite explotar las soluciones mientras el POS es un poco más disperso lo que implica una mayor exploración algo que puede ser una desventaja en los procesos de calibración debido a su alto tiempo computacional. La característica que permite que el AG acote el espacio de soluciones es el cruzamiento ya que cada gen del hijo es obtenido como un aleatorio uniforme entre los genes de los padres. Si a lo anterior se le suma que se realizan dos torneos de tamaño 3 para la selección de los padres de una población de 20 se tiene una probabilidad de 0.27 para la selección del mejor individuo. Una probabilidad grande que

ocasiona que el algoritmo converja de manera rápida y esta probabilidad crece a medida que el esquema del mejor individuo se repite en la población.

5.2.4. Búsqueda realizada por los algoritmos

El mapa de calor mostrado en la Figura 30 es un complemento al análisis de los resultados, ya que permite identificar el valor de los parámetros en cada una de las simulaciones y para cada método. En esta figura se puede notar cuales son los valores que predominan en la búsqueda realizada por los algoritmos y se observa que los parámetros 1 y 2 en cada panel son más oscuros, es decir negativos, mientras que los parámetros 3 y 4 son más claro lo que indica datos positivos.

En la figura también puede apreciarse como los valores encontrados por el algoritmo genético en el panel 22227 tienden a homogenizarse indicando con ello una caída en un óptimo local, en el PSO no se observa tal comportamiento. En el panel 22238 para el AG se nota un fuerte contraste entre valores claros y oscuros indicando con ello que el algoritmo encontró valores buenos para la función objetivo cuando el valor del parámetro 4 o era muy positivo o cerca de cero. En los demás se notan algunos contrastes, debidos seguramente al componente de mutación de este algoritmo.

Por su parte, el PSO es el que en general más contraste muestra de los tres algoritmos lo que induce a pensar que es el que más explora el espacio de soluciones resultado consistente con lo apreciado en el diagrama de coordenadas paralelas.

El SPSA muestra un patrón que oscila entre claro y oscuro para los valores de los parámetros, el cual es consistente debido a que no es propiamente una técnica que permita la exploración sino más bien la explotación de un punto del espacio de soluciones.

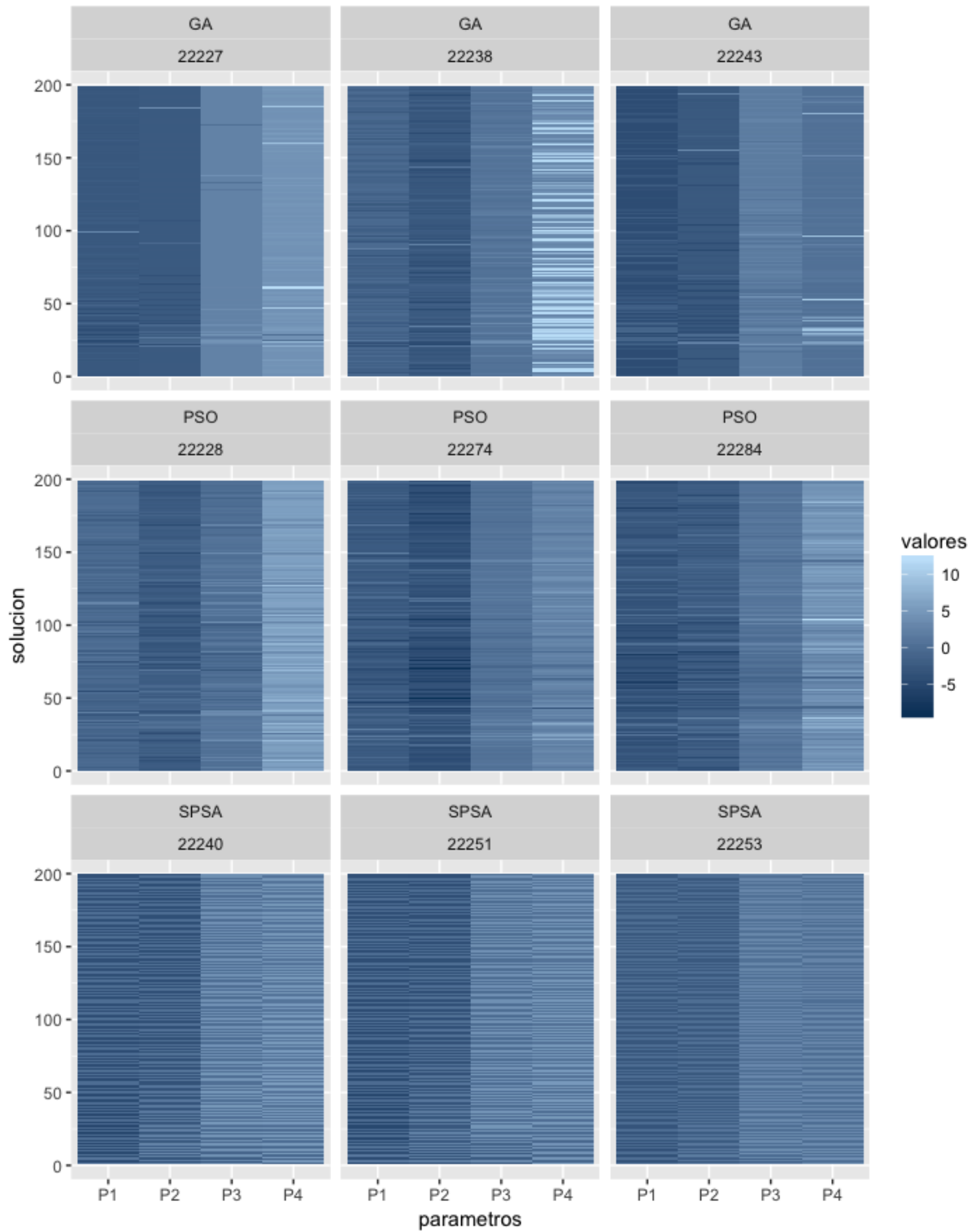


Figura 30. Mapa de calor de los algoritmos implementados. Fuente: Elaboración Propia.

5.3. Gráficos de convergencia

Uno de los intereses que se tiene en calibración es encontrar métodos que converjan rápidamente ya que con ello se puede ahorrar tiempo. En la

Figura 31 se aprecia la convergencia de los algoritmos. En esta no se tiene en cuenta el SPSA debido a que la magnitud del movimiento en dirección al gradiente es apenas significativa debido al coeficiente a_k en la ecuación 3.4.3 y requeriría de simular la nueva posición encontrada, la cual no aporta información al algoritmo y se desperdiciaría tiempo y recursos valiosos (los procesos de calibración llevados a cabo tardan alrededor de 60 horas).

Siguiendo con el análisis, se nota como convergen el AG y el PSO en la búsqueda de soluciones. En el eje x se observa la simulación mientras que, en el eje y el valor de la función objetivo, de tal manera que lo que se observa en esta gráfica es la evolución de los algoritmos buscando soluciones mejores a medida que aumenta en número de simulaciones, aquí sólo se tiene en cuenta la mejor solución encontrada, de tal forma que si las iteraciones van aumentando y la función objetivo no disminuye se presenta una línea horizontal continua.

En la

Figura 31 se observan tres grupos claramente definidos de líneas, se observa que el PSO en algunos procesos de calibración no logró mejorar de manera significativa el GoF (las líneas más altas) esto sugiere que el algoritmo no escapó de

regiones que para el caso son malas. Una gran desventaja ya que se espera que los algoritmos en calibración puedan sortear este tipo de situaciones. El AG presenta un comportamiento estable y se nota como puede explotar de manera exitosa ciertos óptimos. Para continuar con el análisis en la Figura 32 solo se presenta las convergencias para las cuales el GoF sea menor o igual a 10000.

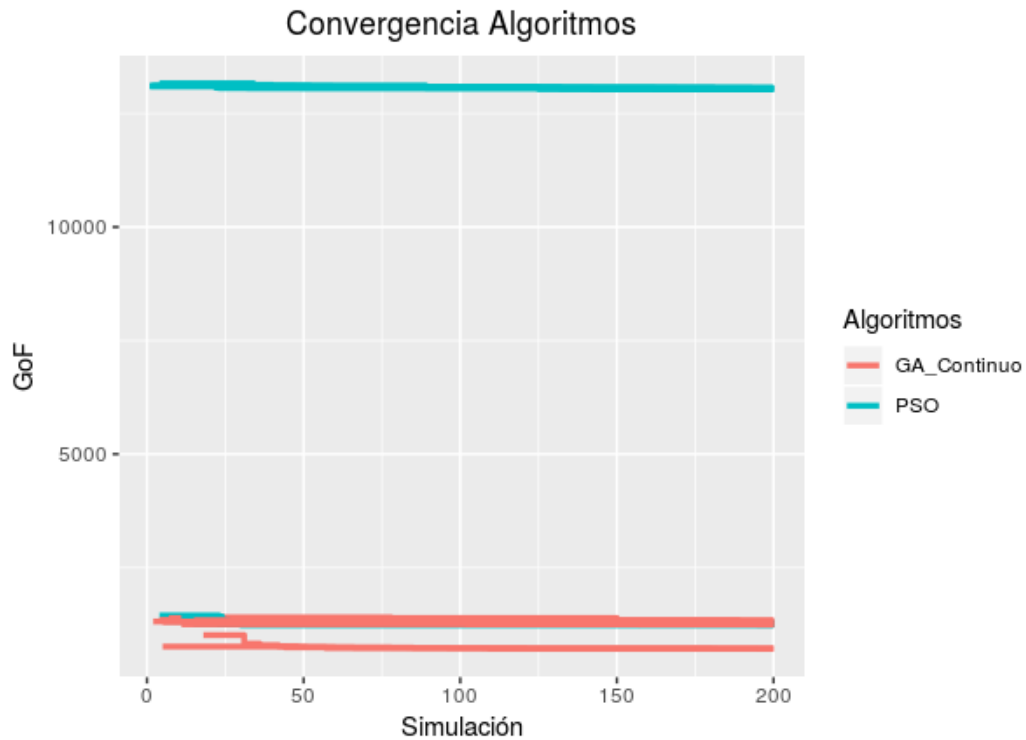


Figura 31. Convergencia de los Algoritmos. Fuente: Elaboración Propia

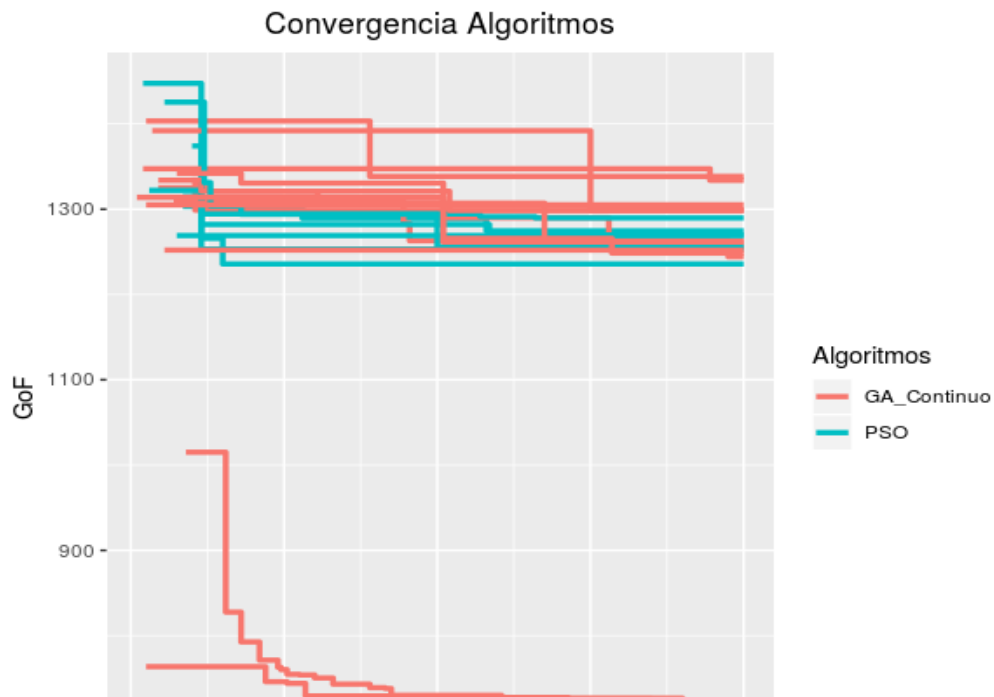


Figura 32. Convergencia de los Algoritmos con Filtro. Fuente: Elaboración Propia.

En la Figura 32 llama la atención dos situaciones. La primera se mencionó anteriormente y es la capacidad del AG para explotar óptimos locales y la segunda es que si la población inicial es similar para el AG y el PSO este último experimenta una convergencia mejor; aunque como se nota en la Figura 31 este no puede escapar de situaciones adversas lo que representa una gran desventaja ya que sugiere que no logra salir de locales muy malos. Frente a tal condición el AG cuenta con un operador de mutación que puede que le puede permitir sobre llevar tal dificultad.

5.4. Evaluación de los resultados.

En este numeral se identifica cuál de los algoritmos presenta mejores resultados para la zona de bajo estudio, considerando la solución encontrada, la convergencia y la búsqueda de las soluciones.

5.4.1. Valores numéricos de los resultados obtenidos.

Luego de culminado el proceso de calibración y del análisis descriptivo de los resultados se tiene en la Tabla 9 los mejores valores encontrados por cada una de las metaheurísticas. En esta se nota como los valores encontrados por el SPSA y el AG son consistentes, varían muy poco unas con otras, mientras el PSO presenta mayor diferencia en los valores encontrados; lo anterior puede deberse a que este algoritmo explora el

espacio de soluciones por lo que puede encontrar varias respuestas, aunque esta exploración impida una explotación que lleve a un mejor ajuste y por ende mejores valores para los parámetros.

Tabla 9. Valores de los resultados.

	Parámetro 1	Parámetro 2	Parámetro 3	Parámetro 4
Algoritmo	-4.37	-2.54	1.68	0.41
Genético	-4.35	-2.54	1.72	0.46
PSO	-3.80	-4.75	1.28	2.01
	-0.32	-2.33	0.32	5.83
SPSA	-1.93	-2.049	2.067	2.681
	-1.954	-2.025	1.968	2.596

5.4.2. Gráficos de densidad.

Con los valores de la Tabla 9, que corresponden como se mencionó con anterioridad a los valores calibrados para los parámetros, se pueden comparar los resultados encontrados por los algoritmos con los valores que por defecto trae el software. Para realizar el proceso de validación se realizan 2000 simulaciones para cada uno de los conjuntos de parámetros con una matriz origen destino diferente a la utilizada en el proceso de calibración, esto con el fin de evitar un sobre ajuste en el modelado, los resultados son mostrados en la Figura 33. Debido a que en la Tabla 9 se presentan dos soluciones para los valores de los parámetros por algoritmo en la Figura 33 y en la Figura 34 se muestran los dos resultados para cada uno de estos conjuntos.

Se observa en la Figura 33 para el PSO y el SPSA mayor dispersión que para el algoritmo genético, así mismo los valores de la función objetivo que se encuentran con el AG son significativamente menores que los demás algoritmos implementados, evidencia de que logró mejores resultados.

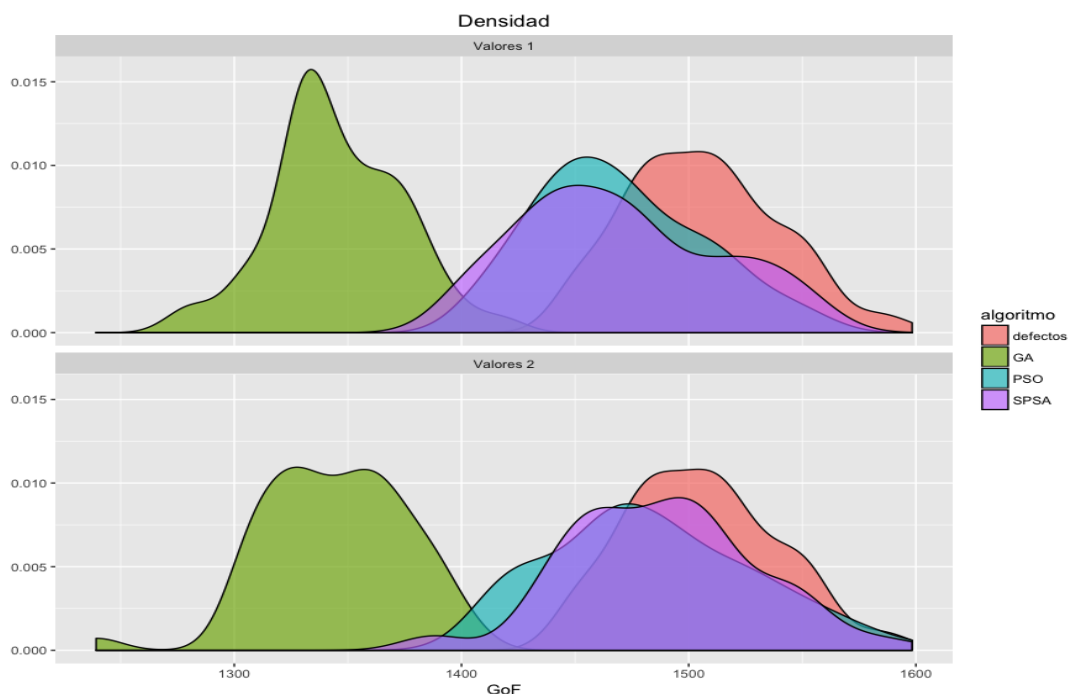


Figura 33. Gráficos de densidad. Fuente: Elaboración Propia.

Analizando los diagramas de box-plot de la Figura 34 se puede notar que el promedio de la función objetivo del PSO y el SPSA es ligeramente mejor que para los valores por defecto identificado como el primer box-plot en rojo. Así mismo, se nota igualmente una mayor dispersión para los valores obtenidos con el PSO y el SPSA Frente al AG, puede notarse que el promedio para el algoritmo genético es el menor.

La Figura 33 y la Figura 34 Figura 34 presentan los resultados más importantes de las investigación ya que en estas se evidencia que los tres algoritmos mejoran el desempeño de los valore que el simulador tiene por defecto configurados. Encontrando valores que no serían posibles sin las técnicas empleadas. Cabe resaltar de igual manera que el AG con configuración real es el que mejor comportamiento exhibe encontrado valores mucho mejores y consistentes que los demás.

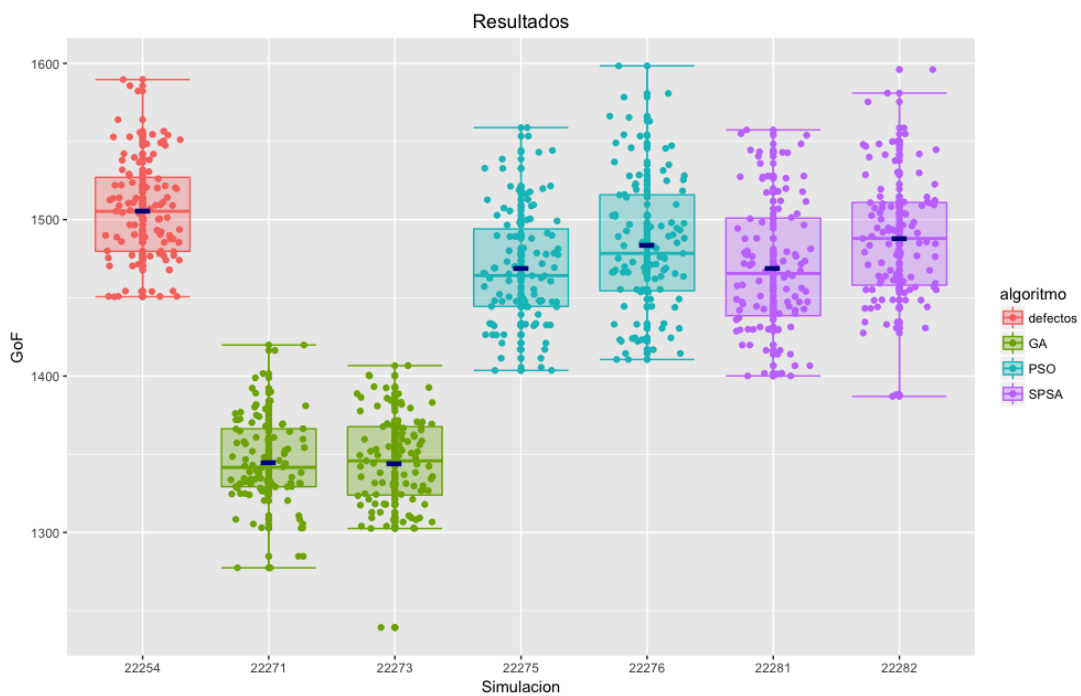


Figura 34. Box plot de los resultados. Fuente: Elaboración Propia.

5.4.3. Gráficos con los intervalos de confianza.

Con los resultados anteriores se puede entonces preguntar si la diferencia de las medias es significativa o no. Es decir, si en promedio, la función objetivo de las metaheurísticas es mejor que el de los valores por defecto; claramente esto ocurre con los algoritmos genéticos no siendo fácilmente apreciable para los dos restantes.

Para identificar si todos los promedios son menores a los valores por defecto se realiza un intervalo de confianza para la media tal como se muestra en la Figura 35, en ella la línea vertical es utilizada para identificar si hay diferencias entre los promedios. Básicamente lo que se tiene es la diferencia de medias de los valores por defecto con los valores encontrados por los algoritmos. Si la línea roja cruza alguna de las barras horizontales de los algoritmos, indicaría que no hay diferencia entre las medias, de lo contrario los valores de los algoritmos son diferentes a los valores por defecto de Transmodeler. Para el caso de estudio todos los valores de los parámetros hallados son diferentes y mejores que los que tienen el software por defecto.

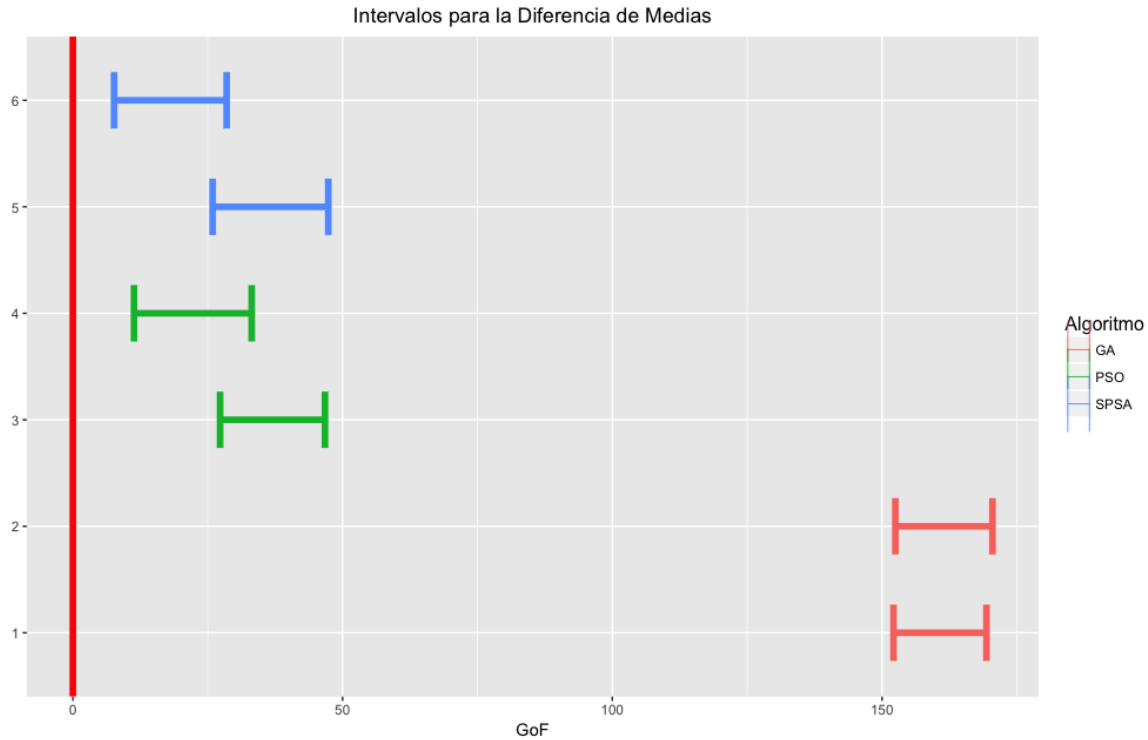


Figura 35. Intervalos para la diferencia de medias. Fuente: Elaboración Propia.

5.4.4. Indicador de convergencia

Un indicador propuesto para evaluar la convergencia consiste en cuantificar el área que dejan los resultados de la Figura 16, tal como se sugiere en (Jiménez, 2016), para los algoritmos implementados de manera individual. La Tabla 10 muestra los valores para el algoritmo genético y el PSO, debido a que como se menciona anteriormente los valores para el SPSA no fueron obtenidos ya que no mostró evidencia de una convergencia rápida y a que requeriría realizar simulaciones de la nueva posición que no aportarían nueva información al desarrollo de la búsqueda.

Tabla 10. Indicador de convergencia.

Algoritmo	Indicador
AG	3423920.0
PSO	14846325

Al realizar la diferencia de medias entre las áreas obtenidas para los algoritmos, se obtiene la Figura 36, se aprecia que la línea roja no cruza la verde lo que implica que el área abarcada por el PSO en las repeticiones fue mayor al AG indicando con esto un mejor comportamiento para el último.

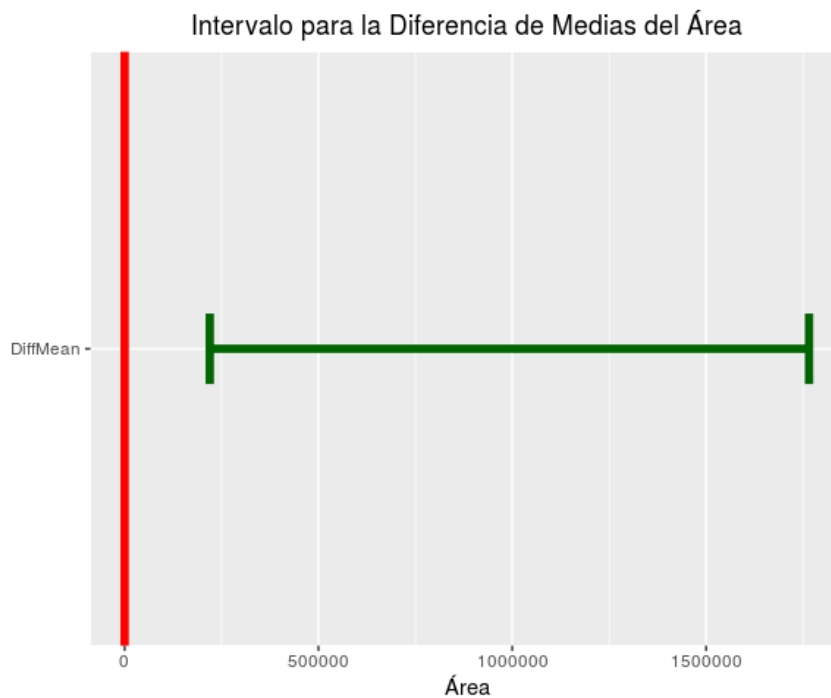


Figura 36. Diferencia de Medias para el Área. Fuente: Elaboración Propia

5.5. Comparación de los resultados con los reportados en la literatura

Los resultados de la presente investigación son comparados con algunos reportados en la literatura los cuales tuvieron un enfoque similar, empezando por el trabajo desarrollado en (Abdalhaq, 2014), allí se comparan los tres algoritmos utilizados en la presente investigación más búsqueda tabú. El algoritmo genético, aunque es real, realizan

un cruzamiento simple y otro en el que se calcula el promedio entre los alelos de los padres. Esa investigación concluye que el PSO presenta un mejor comportamiento que el AG y que es una técnica poco utilizada con mejor comportamiento. Algo contradictorio a la presente propuesta en la que el PSO no encuentra las respuestas halladas por el AG. Lo anterior puede deberse a que como bien argumenta en (Punzo, Vincenzo; Simonelli, 2005) la solución se ve influida por las medidas de rendimiento, la función objetivo y la estrategia de búsqueda.

En general, en la bibliografía se observa que los algoritmos genéticos son los que presentan menor desempeño lo que sugiere que la configuración, la codificación del utilizado en la presente investigación es atractiva para abordar el problema de calibración. Otro punto a favor es que en esta propuesta doce experimentos por algoritmo son llevados a cabo. Cada experimento consiste en un proceso de calibración que da como resultado cada una de las líneas de la Figura 32 esto con el fin de identificar si existe un patrón en la convergencia ya que con una sola no se podría validar estadísticamente.

6. Conclusiones y trabajos futuros

6.1. Conclusiones

La heurística SPSA y las metaheurísticas de AG y PSO, fueron implementadas con éxito en el modelo de microsimulación Transmodeler para la zona del parque Olaya Herrera de la ciudad de Pereira. La primera de ellas es ampliamente abordada en la literatura, para la segunda se realizó una codificación real poco utilizada en el estado del arte, mientras el PSO es una estrategia en la que muy pocos trabajos la han utilizado para afrontar el problema de calibrar un modelo de microsimulación.

Los tres algoritmos utilizados para realizar la calibración encontraron mejores valores para la función objetivo que en promedio son de mejor calidad que los valores que trae el software por defecto; es decir, permiten encontrar los valores de los parámetros que ajustan el modelo a la realidad estudiada.

Por otro lado, debido a que el proceso de calibración consume tiempo, se esperaba que alguno de los algoritmos tuviera una tasa de convergencia mayor, esto con el fin de disminuir el número de simulaciones. Los resultados muestran que el AG puede presentar una mejor convergencia que el PSO, además se le suma la característica de que explota de manera considerable un buen óptimo local. El PSO no logra escapar de regiones de mala calidad algo que presenta una gran desventaja ya que las metaheurísticas deberían sortear este inconveniente. Por último, el SPSA requiere de un ajuste más fino para el valor del a_k que permita mayor movimiento hacia la dirección de descenso.

El algoritmo genético tiende a ser consistente es su búsqueda mostrando poca dispersión lo que implica una que explota el óptimo local encontrado, allí se dirige la

población. La consistencia del SPSA se debe a sus características propias, mientras el PSO mantiene diversidad.

Los parámetros calibrados por el algoritmo genético presentan un valor para la función objetivo menor en promedio que los otros métodos y que los valores por defecto para la zona de estudio.

El problema de optimización basada en simulación para la calibración del modelo de microsimulación es multimodal siendo el algoritmo genético el más apropiado de los métodos explorados, ya que se evidenció que puede mantener varios óptimos locales mientras los explota. El PSO, aunque presenta diversidad no logra respuestas de la calidad que el genético.

El algoritmo genético ofrece la ventaja de que sus parámetros (tasa de mutación, de cruzamiento, tamaño del torneo, tamaño de la población) no requieren de mayor esfuerzo para ser configurados, basta con los valores reportados en la literatura. No ocurriendo lo mismo con el SPSA y el PSO, en estos se requiere de esfuerzo para configurar sus valores. Esto es una enorme desventaja ya que el proceso de calibración por si ya requiere de tiempo y añadir una tarea más al proceso no lo hace eficiente.

La investigación aplicada, genera retos y compromisos significativos puesto que la realidad siempre es más compleja de lo que se piensa. Este trabajo, implementó algoritmos específicos para la calibración en Transmodeler de la zona del parque Olaya Herrera de la ciudad de Pereira; sin embargo, los algoritmos pueden ser utilizados para calibrar una amplia gama de simuladores.

6.2. Trabajos Futuros

Debido a que el proceso de ajuste de los parámetros de los algoritmos consume tiempo y estos valores pueden variar de acuerdo con el lugar de estudio, cantidad y valores a calibrar se propone para este caso generar un metamodelo del problema que permita identificar cuáles valores son buenos para los algoritmos planteados esto permite reducir el tiempo empleado en esta tarea ya que no se requiere de realizar simulaciones.

El parámetro a_k del SPSA influye fuertemente al momento de actualizar la posición por lo que se proponen buscar nuevas formas para estimarlo como la proporción aurea como se argumenta en la literatura. También se propone realizar una combinación entre un PSO o un AG con el SPSA de manera que con los primeros se explore el espacio de soluciones y con el último se explote.

Debido a que el algoritmo genético presentó el mejor comportamiento, algunas variantes de este como los son los algoritmos genéticos basados en selección artificial pueden llegar a mejorar la tasa de convergencia. Adicionalmente, se propone realizar un diseño de experimento factorial completo ya que hay evidencia para afirmar que los parámetros 3 y 4 pueden no aportar a la varianza.

Uno de los inconvenientes que se presentan en los problemas de simulación basada en optimización es la varianza que genera el simulador. Por lo que se considera que una estrategia que utilice información de reducción de varianzas puede llevar a resultados mejores.

El algoritmo genético implementado puede ser adaptado a otras zonas de estudio, y contando con medidas de campo adicionales como velocidades y tiempos de viaje la

presentación del algoritmo trabajado puede enriquecerse con otras funciones objetivo, por ejemplo, incorporando tratamientos multiobjetivo.

En la Tabla 11 se propone un listado de posibles trabajos futuros.

Tabla 11. Resumen de trabajos futuros propuestos

Número	Trabajo futuro propuesto
1	Métodos híbridos para la calibración de un modelo de microsimulación.
2	Solución al problema de calibración usando técnicas multiobjetivo.
3	Metodologías de reducción de varianza para la calibración del modelo de microsimulación.
4	Desarrollo de metamodelos para la calibración de microsimuladores.
5	Desarrollo de métodos numéricos para identificar los mejores valores de los coeficientes del algoritmo SPSA.
6	Inter y extrapolación multidimensional para reducir la cantidad de evaluaciones del SPSA.
7	Algoritmos genéticos basados en la selección artificial para calibrar el modelo de microsimulación.
8	Desarrollo de indicadores para evaluar convergencia de algoritmos en el problema de calibración.

Fuente: Elaboración propia.

Bibliografía

Abdalhaq, B. K., & Baker, M. I. A. (2014). Using Meta Heuristic Algorithms to Improve Traffic Simulation. *Journal of Algorithms and Optimization*, 2(4), 110–128.

- Abdulhai, B., Sheu, J.-B., & Recker, W. (1999). Simulation of ITS on the Irvine FOT Area Using “Paramics 1.5” Scalable Microscopic Traffic Simulator: Phase I: Model Calibration and Validation. Partners for Advanced Transit and Highways (PATH), (April 2016). Retrieved from <http://escholarship.org/uc/item/2ks86938#page-1>
- Alcaldía Pereira. Cable Aéreo Línea Parque Olaya-Terminal-UTP-Villa Santana, Pub. L. No. 271, 7 (2017). Colombia: <http://pereira.gov.co/Transparencia/Normatividad/DECRETOS/2017/Decreto%20271%20del%2030%20de%20marzo%20de%202017%20Proyecto%20Cable%20A%C3%A9reo.pdf>.
- Antoniou, C., Barcelo, J., Brackstone, M., Celikoglu, H. B., Ciuffo, B., Punzo, V., ... Wagner, P. (2014). Traffic Simulation: Case for guidelines. <https://doi.org/10.2788/11382>
- Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H., & Wen, Y. (2007). Calibration of Microscopic Traffic Simulation Models: Methods and Application. *Transportation Research Record: Journal of the Transportation Research Board*, 1999, 198–207. <https://doi.org/10.3141/1999-21>
- Banco Mundial. (2017). Movilidad. Retrieved June 7, 2018, from <http://www.bancomundial.org/es/results/2017/12/01/mobility>
- Barceló, J. (2010). *Fundamentals of Traffic Simulation* (Vol. 145). <https://doi.org/10.1007/978-1-4419-6142-6>
- Brockfeld, E., Kühne, R. D., & Wagner, P. (2004). Calibration and validation of microscopic traffic flow models. *Transportation Annual Meeting 2004*, (750).
- Caliper. (2011). *TransModeler Basics Introduction to TransModeler* (pp. 1–593).
- Caliper. (2018). *TransModeler Traffic Simulation Software*. Retrieved July 23, 2018, from <https://www.caliper.com/transmodeler/default.htm>
- Carson, Y., & Maria, A. (1997). Simulation Optimization: Methods and Applications. *Proceedings of the 29th Conference on Winter Simulation* (1997), 118–126. <https://doi.org/10.1145/268437.268460>
- Chu, L., & Liu, H. X. (2004). A Calibration Procedure for Microscopic Traffic Simulation. In *TRB Annual Meeting* (pp. 1–25).
- Ciuffo, B., & Punzo, V. (2014). “No free lunch” Theorems applied to the calibration of

traffic simulation models. *IEEE Transactions on Intelligent Transportation Systems*, 15(2), 553–562. <https://doi.org/10.1109/TITS.2013.2287720>

Ciuffo, B., Punzo, V., & Montanino, M. (2012a). The calibration of traffic simulation models: Report on the assessment of different goodness of fit measures and optimization algorithms. *JCR Scientific and Technical Reports*. <https://doi.org/10.2788/7975>

Ciuffo, B., Punzo, V., & Montanino, M. (2012b). The calibration of traffic simulation models: Report on the assessment of different goodness of fit measures and optimization algorithms.

Ciuffo, B., Punzo, V., & Montanino, M. (2012c). The Calibration of Traffic Simulation Models Report on the assessment of different Goodness of Fit measures and Optimization Algorithms MULTITUDE Project – COST Action TU0903. *JCR Scientific and Technical Reports*. <https://doi.org/10.2788/7975>

Cobos, C., Erazo, C., Luna, J., Mendoza, M., Gaviria, C., Arteaga, C., & Paz, A. (2016). *Advances in Artificial Intelligence*. CAEPIA, 9868, 468–476.

Daamen, W., Buisson, C., & Hoogendoorn, S. P. (2014a). *Traffic Simulation and Data*. (W. Daamen, C. Buisson, & S. P. Hoogendoorn, Eds.). New York: CRC Press Taylor & Francis Group.

Daamen, W., Buisson, C., & Hoogendoorn, S. P. (2014b). *Traffic Simulation and Data*.

Daigle, G., Thomas, M., & Vasudevan, M. (1998). Field Applications of CORSIM: I-40 Freeway design evaluation, Oklahoma City. *Proceedings of the 1998 Winter Simulation Conference*, 1161–1167.

DANE. (2018). Reloj de población. Retrieved June 7, 2018, from <http://www.dane.gov.co/reloj/>

Dowling, R., Skabardonis, A., & Alexiadis, V. (2004). *Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software*. Rep. No. FHWA-HRT-04-040, U.S. DOT, Federal Highway Administration, Washington, D.C (Vol. III).

Dowling, R., Skabardonis, A., Halkias, J., McHale, G., & Zammit, G. (2004). Guidelines for Calibration of Microsimulation Models: Framework and Applications. *Transportation Research Record: Journal of the Transportation Research Board*, 1876(1876), 1–9. <https://doi.org/10.3141/1876-01>

- Eiben, A. ., & Smith, J. . (2008). Introduction to evolutionary computing. (S. Editors, R. Th, E. J. N. K. H. P. Spaink, A. Board, A. G. Brassard, K. a D. J. C. C. a M. Gielen, ... D. W. E. Winfree, Eds.), *ACM Computing Classification* (Vol. 1). <https://doi.org/10.1162/evco.2004.12.2.269>
- Fu, M. C. (2013). Simulation Optimization. In S. I. Gass & M. C. Fu (Eds.), *Encyclopedia of Operations Research and Management Science* (pp. 1418–1423). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4419-1153-7_958
- Gardes, Y., May, A. D., Dahlgren, J., & Skabardois, A. (2002). Freeway Calibration and Application of the Paramics Model. In 81st Annual Meeting of the Transportation Research Board, Washington, DC., (January), 0–22.
- Goldberg, D. (1991). Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, 5, 139–167. <https://doi.org/10.1.1.52.9880>
- Herrera, F., Lozano, M., & Verdegay, J. L. (1998). Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Artificial Intelligence Review*, 12(4), 265–319. <https://doi.org/10.1023/A:1006504901164>
- Holland, J. H. (1992). *Systems, Adaptation in Natural and Artificial* (2nd Editio). MIT Press Cambridge, MA, USA.
- Hollander, Y., & Liu, R. (2008). The principles of calibrating traffic microsimulation models. *Transportation*, 35(3), 347–362. <https://doi.org/10.1007/s11116-007-9156-2>
- Huang, W., & Sun, J. (2009). A NGSa - II based parameter calibration algorithm for traffic microsimulation model. *International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2009*, 3(2), 436–439. <https://doi.org/10.1109/ICMTMA.2009.437>
- Huang, W., Sun, J., Wang, H., & Li, K. (2008). Design of IDSS for Traffic Microsimulation Model Calibration. *2008 Second International Symposium on Intelligent Information Technology Application*, 1, 550–555. <https://doi.org/10.1109/IITA.2008.229>
- Jiménez, A. (2016). Uso de técnicas estadísticas para la construcción de funciones objetivos en el proceso de calibración de modelos de simulación de tráfico.
- Jiménez, D., Muñoz, F., Arias, S., & Hincapie, J. (2016). Software for calibration of transmodeller traffic microsimulation models. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 1317–1323.

<https://doi.org/10.1109/ITSC.2016.7795727>

- Kim, K.-O., Rilett, L. R., & Kim, I. (2001). Genetic-Algorithm-based Approach for Calibrating Microscopic Simulation Models. *IEEE Intelligent Transportation Systems Conference Proceedings -Oakland*.
- Lee, J., & Ozbay, K. (2008). Calibration of a Macroscopic Traffic Simulation Model Using Enhanced Simultaneous Perturbation Stochastic Approximation Methodology. *Transportation Research Record*, 1–19.
- Lidbe, A. D., Hainen, A. M., & Jones, S. L. (2017). Comparative study of simulated annealing, tabu search, and the genetic algorithm for calibration of the microsimulation model. *Simulation*, 93(1), 21–33. <https://doi.org/10.1177/0037549716683028>
- Ma, J., Dong, H., & Zhang, H. (2007). Calibration of Microsimulation with Heuristic Optimization Methods. *Transportation Research Record: Journal of the Transportation Research Board*, 1999, 208–217. <https://doi.org/10.3141/1999-22>
- Ma, T., & Abdulhai, B. (2001). Genetic algorithm-based combinatorial parametric optimization for the calibration of microscopic traffic simulation models. *IEEE Intelligent Transportation Systems Conference Proceedings*, 848–853. <https://doi.org/10.1109/ITSC.2001.948771>
- Marini, F., & Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 149, 153–165. <https://doi.org/10.1016/j.chemolab.2015.08.020>
- Mengshoel, O. J., Galán, S. F., & De Dios, A. (2014). Adaptive generalized crowding for genetic algorithms. *Information Sciences*, 258, 140–159. <https://doi.org/10.1016/j.ins.2013.08.056>
- Naciones Unidas. (2018). Población | Naciones Unidas. Retrieved March 26, 2018, from <http://www.un.org/es/sections/issues-depth/population/index.html>
- Paz, A., Molano, V., Martinez, E., Gaviria, C., & Arteaga, C. (2015). Calibration of traffic flow models using a memetic algorithm. *Transportation Research Part C: Emerging Technologies*, 55(4), 432–443. <https://doi.org/10.1016/j.trc.2015.03.001>
- PTV Group. (2018). Detailed Traffic Simulation Software. Retrieved July 23, 2018, from <https://www.ptvgroup.com/en/solutions/products/ptv-vissim/>

- Punzo, Vincenzo; Simonelli, F. (2005). Analysis and comparison of microscopic traffic flow models with real traffic microscopic data. *Transportation Research Board: Journal of the Transportation Research Board.*, No. 1934, 53–63. <https://doi.org/10.3141/1934-06>
- Robledo, S., Osorio, G. A., & López, C. (2014). Networking en pequeña empresa: una revisión bibliográfica utilizando la teoría de grafos. *Revista Vínculos*, 11(2), 6–16.
- Rodrigues, R. (2014). *Transport Modelling : Macro and Micro Simulation for the Studied Case of Funchal*. University of Madeira.
- Sosa, A., Kogan, J., Azán, S., Miquilena, M., & Alcántra, E. (2011). Desarrollo urbano y movilidad en América Latina. Banco de desarrollo de America Latina CAF. <https://doi.org/10.1109/MIC.2013.83>
- Spall, J. (2003a). *Introduction to stochastic search and optimization*. Wiley-Interscience, New York.
- Spall, J. (2003b). *Introduction to stochastic search and optimization*. (WILEY, Ed.).
- Spall, J. C. (1998). Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems*. <https://doi.org/10.1109/7.705889>
- Toledo, T., Ben-Akiva, M., Darda, D., Jha, M., & Koutsopoulos, H. (2004). Calibration of Microscopic Traffic Simulation Models with Aggregate Data. *Transportation Research Record: Journal of the Transportation Research Board*, 1876, 10–19. <https://doi.org/10.3141/1876-02>
- Toledo, T., Koutsopoulos, H. N., Davol, A., Ben-Akiva, M. E., Burghout, W., Andréasson, I., ... Lundin, C. (2003). Calibration and Validation of Microscopic Traffic Simulation Tools: Stockholm Case Study. *Transportation Research Record*, (03), 65–75. Retrieved from <http://www.scopus.com/inward/record.url?eid=2-s2.0-1642312081&partnerID=40&md5=4005730df8f72ee1095e85432daac4e8>
- Treiber, M., & Kesting, A. (2013). *Traffic Flow Dynamics*. Springer. <https://doi.org/10.1007/978-3-642-32460-4>
- Wackerly, D. D., Mendenhall, W., & Schaffer, R. L. (2009). *Estadística matemática con aplicaciones*. Cengage learning.
- Whitley, D., & Kauth, J. (1988). GENITOR: A different genetic algorithm. In *Rocky*

Mountain Conference on Artificial Intelligence (pp. 118–130).

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>

Wu, Z., Sun, J., & Yang, X. (2005). Calibration of VISSIM for Shanghai Expressway using genetic algorithm. *Proceedings - Winter Simulation Conference, 2005*, 2645–2648. <https://doi.org/10.1109/WSC.2005.1574564>

Xinjie, Y., & Mitsuo, G. (2010). *Introduction to Evolutionary Algorithms*. Springer-Verlag London. <https://doi.org/10.1017/CBO9781107415324.004>

Yu, M., & (David) Fan, W. (2017). Calibration of microscopic traffic simulation models using metaheuristic algorithms. *International Journal of Transportation Science and Technology*, 6(1), 63–77. <https://doi.org/10.1016/j.ijtst.2017.05.001>

Anexos

Este trabajo contempla los siguientes anexos que son parte integral del documento:

Anexo_1_ Búsquedas_Indexadas.

Anexo_2_Estado_del_Arte.