OPTIMIZACIÓN EN LA DISTRIBUCIÓN DE RUTAS DE COLEGIOS PRIVADOS EN PEREIRA CON AYUDA DE ALGORITMOS DE INTELIGENCIA ARTIFICIAL

STEVEN PINEDA CORTÉS
RAFAEL PINZÓN RIVERA

UNIVERSIDAD TECNOLÓGICA
DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
MAYO DE 2015

OPTIMIZACIÓN EN LA DISTRIBUCIÓN DE RUTAS DE COLEGIOS PRIVADOS EN PEREIRA CON AYUDA DE ALGORITMOS DE INTELIGENCIA ARTIFICIAL

STEVEN PINEDA CORTÉS
RAFAEL PINZÓN RIVERA

INFORME DE PROYECTO DE GRADO DE PREGRADO

UNIVERSIDAD TECNOLÓGICA
DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
MAYO DE 2015

Agradecimientos

A Julio Hernando Vargas que nos apoyó en el desarrollo de todo este proyecto, así también como a nuestras familias que con su apoyo fue posible todo el desarrollo este proyecto.
También a la Universidad Tecnológica de Pereira, que nos brindó muchos de los conocimientos necesarios para la realización del proyecto.

Índice general

Índice de figuras

Índice de cuadros

Glosario

1. VRP: Vehicule Routing Problem o Problema de enrutamiento de vehículos es un problema de optimización combinatoria con diferentes variaciones que busca distribuir de la mejor manera un conjunto de rutas que deben atender a un conjunto de usuarios.

2. IA: Inteligencia Artificial. Donde se le atribuye inteligencia a un programa de cómputo, siendo este capaz de interactuar con su entorno.

3. Algoritmos genéticos: Conjunto de pasos que buscan encontrar solución a un problema basándose en la selección natural de los organismos, simulando la selección, reproducción y mutación a través de generaciones.

4. Simulated annealing: En español enfriamiento simulado, es un algoritmo que imita un proceso metalúrgico para alcanzar el mínimo nivel de energía.

Resumen

En este proyecto se realizó una investigación de los distintos algoritmos utilizados para resolver el problema de enrutamiento de vehículos, VRP, con el fin de facilitar esta tarea a los encargados de rutas en los colegios privados de la ciudad de Pereira. De acuerdo a los resultados encontrados y al conocimiento en algoritmos se eligieron dos para llevarlos a una aplicación con la que se puede optimizar la distribución de rutas para un mapa, un conjunto de estudiantes y unas rutas dadas, minimizando los recorridos basados en la distancia geográfica entre los estudiantes. Los algoritmos de inteligencia artificial usados fueron el Genético y el Recocido Simulado (Simulated Annealing).
Se realizaron pruebas con dos colegios privados, el Liceo Taller San Miguel y el Saint Andrews, ubicados en la vía a Armenia y en la vía a Cerritos respectivamente, las dos zonas con mayor concentración de colegios privados de la ciudad. Se identificaron las mejores configuraciones y combinación de los algoritmos dados y se pudieron observar resultados satisfactorios en la medida en que no se logran identificar mejoras.

Capítulo 1

Introducción

1.1. Descripción del Problema

Una gran cantidad de colegios, que incluye casi la totalidad de los privados de Pereira, prestan el servicio de transporte a sus estudiantes. Estos colegios pueden llegar a tener más de mil estudiantes cada uno y por consiguiente más de 30 buses o microbuses, con diferentes capacidades, para llevar a sus estudiantes.
En general los colegios empiezan con pocos estudiantes y pueden repartir a las personas en diferentes rutas diseñadas manualmente, pero a medida que crece el número de estudiantes este problema se vuelve más complejo y es muy probable que las decisiones tomadas estén lejos de ser las óptimas.
Este tipo de problema se conoce en el mundo de la computación como VRP [34](por sus siglas en inglés Vehicule Routing Problem) o problema de enrutamiento de vehículos. En este problema se dispone de un número finito de recursos o rutas para repartir bienes que ocupan una determinada capacidad de los recursos, los bienes deben repartirse en diferentes puntos y el objetivo es minimizar los costos de repartición, que en general consisten en la suma de las distancias recorridas por cada ruta.
El VRP puede ser representado usando un grafo, donde los vértices incluyen la ubicación de los bienes y del punto de inicio y final, las aristas son los recorridos entre cada vértice y sus costos están determinados por la distancia entre ellos, normalmente todas las conexiones son posibles. También es común usar una matriz donde se relacione la distancia de todos los nodos con cada uno de los demás.
La importancia de abordar el problema con inteligencia artificial radica en la complejidad computacional del problema. Si se hiciera una búsqueda explorando cada una de las soluciones para determinar la mejor, la cantidad de soluciones a explorar estaría determinada así, siendo n el total de alumnos, m el total rutas y r la capacidad de las rutas, asumiendo que la capacidad es igual para todas las rutas y que $m * r >= n$:
Para escoger los alumnos de una primera ruta se tendrían C_r^n posibildades para el segundo serían $C_r^{(n-r)}$, luego $C_r^{(n-2r)}$ y cada uno de estos números se debe multiplicar hasta que no quede ningún estudiante por repartir, por lo cual el resultado final sería:

$$\prod_{i=0}^{m-1} C_r^{(n-i*r)}$$

Que expandiendo la combinatoria es equivalente a:

$$\prod_{i=0}^{m-1} \frac{(n - i * r)!}{r!(n - (i + 1) * r)!} \tag{1.1}$$

Para dar una mejor idea de la cantidad tan amplia de soluciones se hizo el cálculo con algunos valores, observar Cuadro 1.1.

Alumnos	Capacidad de rutas	Posibilidades
20	10	184.756
50	10	4,833E+31
100	20	1,095E+66
200	20	1,086E+191
210	30	1,145E+171

Cuadro 1.1: Posibilidades de distribución de rutas

Como se puede observar en un colegio de 200 estudiantes no tiene sentido explorar todas las soluciones para determinar la óptima, además en Pereira hay colegios con más de 800 estudiantes como el Salesiano Juan Bosco [5], el Liceo Taller San Miguel [6] con alrededor de 700 y La Salle [27] con más de 1000.
Debido a la cantidad de información que se debería analizar la complejidad computacional es bastante alta, por lo que se deben utilizar heurísticas que aunque no garanticen la mejor solución pueden encontrar una buena en un tiempo prudente independientemente de que no sea óptima.

1.1.1. Definición del Problema

El proceso de repartir a los estudiantes en rutas se hace hoy en día de manera empírica y manual en los colegios privados de la ciudad de Pereira, lo que conlleva a planes de distribución difíciles de crear y demanda horas de trabajo, estos planes pueden estar bastante lejos de lo óptimo, causando gastos innecesarios de combustible y tiempo afectando conductores y estudiantes.

1.2. Justificación del proyecto

Una mala distribución en las rutas ocasiona pérdidas de tiempo en estudiantes y conductores, además de generar gastos de gasolina que podrían ser evitados. De otro lado se tiene a las personas encargadas de esta tarea que si lo hacen manualmente les puede tomar horas y se debe hacer cada que inicia un nuevo año escolar o cuando ingresan nuevos estudiantes en cualquier otro momento del año.
Teniendo un software que haga esto automáticamente basado en la ubicación de cada estudiante, el número de rutas y la capacidad de las mismas, se puede encontrar una solución mucho mejor y en menos tiempo que si esta tarea se hace manualmente.
Se espera también aportar a futuras investigaciones o desarrollos en el área de VRP con los datos recopilados y puntos que se deban tener en cuenta para obtener mejores resultados.

1.3. Objetivos

1.3.1. Objetivo General

Determinar un algoritmo que mejor se adapte para el problema de repartición de rutas de colegios en Pereira para desarrollar un software prototipo que realice esta tarea.

1.3.2. Objetivos Específicos

1. Analizar alternativas con que se ha abordado el problema en el pasado.
2. Proponer una nueva alternativa o variación de un algoritmo existente.
3. Comparar alternativas y elegir la más adecuada teniendo en cuenta la distribución geográfica de los colegios objetivo.
4. Diseñar e implementar una aplicación que proponga un plan de distribución de rutas para colegios en la ciudad de Pereira.

1.4. Antecedentes

El problema de distribución de rutas se puede abordar con 3 diferentes tipos de algoritmos; los primeros buscan encontrar una solución exacta pero tienen limitación en el tamaño de problema que pueden resolver, también están los algoritmos que utilizan heurísticas para encontrar una solución buena y con mayor capacidad en cuanto a puntos a repartir, y finalmente están los métodos clasificados como meta heurísticos en donde la búsqueda es más aleatoria pero que se adapta a más problemas y puede producir muy buenas soluciones. Una comparación de estas familias se puede encontrar en [1], [18] y [30].

1.4.1. Algoritmos Exactos

El primer acercamiento a resolver el VRP son los algoritmos exactos, es decir, que garantizan una solución óptima. Estos sin embargo se caracterizan por la limitación en la cantidad de puntos a distribuir que puede resolver y según el estudio de Laporte y Nobert [19] se clasifican en 3 grupos los cuales son: búsqueda de árbol directa, programación dinámica y programación lineal entera.

Búsqueda de árbol directa

La alternativa de búsqueda de árbol directa con mayor alcance conocida fue propuesta por Laporte, Mercure y Norbert en 1987 [19], en la cual se aprovecha la similitud del problema con el MSTP usando un algoritmo branch and bound que consiste en ramificar y podar aquellas soluciones que no llevarán a un camino óptimo. Este algoritmo demostró poder resolver problemas de hasta 250 vértices.

Programación dinámica

En cuanto al uso de programación dinámica para VRP, esta fue propuesta por Eilon, Watson-Gandy y Christofides en 1971 [10] el cual inicialmente podía encontrar una solución exacta para problemas entre 10 y 15 puntos. Posteriormente Christofides hizo algunas mejoras que permitieron que trabajara hasta con 50. Usos más actuales de este algoritmo logran resolver instancias de VRP de hasta 101 vértices en un tiempo de alrededor de 6 horas [20].

Programación lineal entera

En el último grupo se tiene un método de particiones de conjuntos propuesto por Balinski, y Quandt en 1964 [28]; desafortunadamente este solo tiene alcance de hasta 15 vértices e incluso no garantiza una solución exacta siempre. Este procedimiento fue adaptado por Desrosiers y Solomon en 1991 [24] el cual sirve para resolver VRPTW, una modalidad que agrega ventanas de tiempo (las siglas TW son por Time Windows en inglés) en las que los bienes se pueden dejar o recoger con restricciones de tiempo. Este funciona con problemas de hasta 100 lugares y su desempeño mejora a medida que las restricciones aumentan, ya que el campo de soluciones es menor.

1.4.2. Algoritmos Heurísticos

Debido a las limitaciones de los algoritmos exactos se han propuesto múltiples aproximaciones que en general garantizan soluciones buenas, aunque no las óptimas. Estos algoritmos se dividen en 3 familias: constructivas, de dos fases y de mejora.

Constructivas

Las heurísticas constructivas crean soluciones desde cero. Una de ellas y tal vez la más fácil de implementar es la del vecino más cercano, en esta se escoge un punto al azar y se conecta con el que se encuentre a menor distancia aún no conectado, este proceso se repite hasta que todos estén unidos pero en general sus resultados están lejos de los óptimos por lo cual no es muy usado. El algoritmo heurístico más representativo de esta familia es el Clark Wright Savings, propuesto por Clarke y Wright en 1964 [17] en el cual inicialmente se tiene un ruta para cada punto y estas se van combinando de acuerdo a la que produzca un mayor ahorro de recorrido.

Dos fases

Los algoritmos de dos fases se dividen en dos sub-familias: una en la que primero se agrupa y luego se enruta, y otra en la que el procedimiento se hace al revés. Del primer grupo está el algoritmo de barrido (The sweep algorithm) que fue inicialmente sugerido por Wren en 1971, luego por Wren y Holliday en 1972 [2] donde se usaba para CVRP, variante en la que se tiene en cuenta la capacidad de cada ruta y el espacio que ocupa cada bien que se transporta. El algoritmo se puede adaptar a situaciones con uno o múltiples origines de las rutas. Finalmente Gillett and Miller en 1974 [13] le dieron el nombre de algoritmo de barrido y este se popularizó.
Este acercamiento ha probado proveer soluciones entre un 2 % y un 10 % de cercanidad con la solución óptima conocida en problemas de la vida real [29] [34].

Otros métodos de dos fases y agrupamiento primero incluyen al de Fisher y Jaikumars de 1981 [26] el cual usa un método más sofisticado de agrupamiento, resolviéndolo como un GAP o problema de asignación general (General Assignment Process), este logra soluciones hasta un 2 % mejores que las del algoritmo de barrido [34] pero tiene la limitación de que la cantidad de rutas debe ser fijada desde el inicio. También hay una extensión del método de barrido propuesta por Balinski y Quandt y mejorada por Foster y Ryan [7]; esta se conoce como algoritmo de pétalo y su variación consiste en que se produce una colección de rutas candidatas llamadas pétalos, las cuales inicialmente se intersectan y posteriormente se dividen en soluciones posibles, al igual que en las demás cada ruta se soluciona como un TSP aislado.
Los métodos de la segunda sub-familia consisten en primero crear una especie de TSP que agrupe a todos los vértices y luego este se empieza a dividir en rutas, sin embargo no son competitivos con los algoritmos que tienen el orden inverso.

De mejora

La última familia es de los algoritmos de mejora. Uno de los primeros métodos de mejora iterativa fue el 2-Opt, el cual consiste en recorrer parejas de nodos conectados y reordenarlos si esto causa una mejora en el total del recorrido. Una mejora propuesta por Christofides y Eilon [31] fue el 3-Opt, que funciona de igual manera pero se toman de a 3 nodos para intentar buscar un orden más conveniente. Este tipo de heurísticas al limitarse a la mejora de soluciones normalmente se utiliza en combinación con otros algoritmos, usándose como una mejora al resultado obtenido pero nunca para crear soluciones desde cero.

1.4.3. Meta Heurísticas

Las meta heurísticas son métodos de nivel superior a los heurísticos en los que generalmente se crea una solución o conjunto de soluciones, a veces aleatorias, que son mejorados iterativamente en busca del óptimo global. Este tipo de algoritmos suelen aplicarse solamente a problemas que no tienen una buena solución con las heurísticas conocidas debido a que no garantizan que se encuentre incluso una buena solución y habitualmente tienen un margen de efectividad menor, sin embargo en cuanto a VRP algunos de los mejores resultados han sido encontrados con meta heurísticas. El éxito o fracaso depende de la forma en que se implemente y se ajusten las variables del algoritmo; los ajustes se hacen por lo regular de manera empírica. En este tipo de métodos se destacan las colonias de hormigas, la búsqueda Tabú, el enfriamiento simulado (mejor conocido como Simulated Annealing en inglés) y los algoritmos genéticos.

Enfriamiento simulado

El enfriamiento simulado, SA por sus siglas en inglés, está inspirado en un proceso utilizado en la metalúrgica en el cual se calienta un material y se deja enfriar controladamente para reducir sus defectos. Fue propuesto por Scott Kirkpatrick, C. Daniel Gelatt y Mario P. Vecch [8] para resolver problemas de múltiples variables y combinatorios. Lo interesante es que el algoritmo tiene la característica de explorar estocásticamente soluciones cercanas a una solución dada aunque no sean mejores

que la actual, con el fin de escapar a mínimos locales. Las soluciones alternas se aceptan con una probabilidad que depende de una manera directamente proporcional del parámetro global T el cual va disminuyendo con las iteraciones para que converja finalmente. En la analogía con el proceso metalúrgico T representa la temperatura. Este procedimiento se ha utilizado principalmente para resolver VRPTW [4] [23].

Colonia de hormigas

La optimización de colonia de hormigas está basada en la forma en que estos insectos buscan y recolectan comida y fue propuesta por primera vez para el VRP en 1997 por Bullnheimer, Hartl y Strauss [15]. En la vida real las hormigas inicialmente buscan la comida siguiendo caminos aleatorios y una vez la encuentran regresan a su colonia dejando un rastro de feromonas, así cuando otras hormigas salgan en la búsqueda seguirán los rastros que encuentren con una probabilidad directamente proporcional a que tan fuerte es aún la feromona, lo cual les indica hace cuánto tiempo fue depositada. Si existen varios caminos, los más cortos tendrán feromonas más recientes que serán preferidas por otras hormigas, además; si estos insectos siguen el rastro y encuentran la comida repetirán el proceso de volver a su colonia dejando a su paso más feromonas dando como resultado un buen camino que las demás recorrerán.
Esta heurística se aplicó inicialmente a TSP y fue adaptada a VRP por Bullnheimer, Hartl y Strauss forzando a las hormigas a crear una nueva ruta cada vez que excedía su capacidad o una distancia máxima predefinida. Este algoritmo ha probado encontrar soluciones de hasta un 1 % bajo la óptima conocida [11] y ha sido adaptado para resolver VRPTW [14] VRP dinámico o DVRP [12], variante en que las rutas deben ser actualizadas en tiempo real para abarcar nuevos nodos.

Búsqueda Tabú

El concepto de búsqueda tabú fue descrito inicialmente por Glover [16] en 1986 y consiste en que cada iteración una solución se mueve hacia la mejor solución de un subconjunto de su vecindario. Para evitar ciclos, las soluciones tienen información sobre las recientemente exploradas, las cuales son declaradas prohibidas o tabú. La duración de una solución como tabú es determinada por parámetros que pueden variar a través del tiempo y adicionalmente se puede salir de este estatus si por ejemplo una declarada tabú es mejor que cualquier otra encontrada hasta ese momento. La búsqueda tabú inicia con una solución candidata, la cual puede ser generada aleatoriamente o partir de otra heurística.
Entre las versiones de este algoritmo se encuentra el Tabú granular propuesta por Toth y Vigo [33] donde se hace un proceso adicional que elimina movimientos en el vecindario que probablemente no producirán buenos resultados y solo los reintegran si el algoritmo se atasca en un mínimo local. También está el proceso de memoria adaptativa de Rochat y Taillard [9] donde se tiene un conjunto de buenas soluciones que es dinámicamente actualizado a través del proceso de búsqueda y de manera periódica algunos elementos son extraídos y combinados con otros para producir nuevas soluciones buenas, finalmente el de Kelly y Xu que utiliza el intercambio de vértices entre dos rutas, un posicionamiento global de algunos vértices en otras rutas y mejoras locales en cada recorrido [32].

Algoritmos genéticos

Los algoritmos genéticos son uno de los meta heurísticos más conocidos; estos simulan el proceso de selección natural y lo que pasa a nivel de genética para evolucionar hacia mejores soluciones. La aplicación a problemas computacionales fue propuesta por Holland en 1975 [21] y ha sido aplicada a tipos de problemas muy variados. En este método cada solución debe estar codificada en cromosomas que representan al individuo y pueden ser evaluados por una función conocida como fitness, que indica que tan bueno es. Sobre la población o poblaciones de individuos se simula tanto la reproducción como la mutación, donde las soluciones más aptas tienen más posibilidades de reproducirse. Al final el individuo que mejor resultado tenga en la función fitness es decodificado y corresponde a la solución. En todo el proceso hay 3 operaciones importantes, de su implementación y de la representación del problema en cromosomas depende encontrar o no una buena solución, estas son la selección, reproducción y mutación.
Al implementar las operaciones se debe buscar que se dé oportunidad de diversidad en las soluciones, de no ser así el algoritmo puede fácilmente converger en mínimos locales.
Los resultados obtenidos con esta meta heurística no han sido los mejores; sin embargo una adaptación de Nagata de EAX [35] y un algoritmo genético híbrido propuesto por Berger y Barkoui conocido como HGA-VRP (Hybrid Genetic Algorithm – VRP) [25] han alcanzado las mejores soluciones conocidas para instancias clásicas de VRP y son competitivas con los mejores resultados conocidos.

1.4.4. Factor geográfico y colegios

Pereira es la capital del departamento de Risaralda, Colombia. Se ubica en la región central del país en la cordillera central de los Andes, es la más poblada del eje cafetero con más de medio millón de habitantes. El área municipal es de $702km^2$. La cercanía con el municipio vecino, Dosquebradas, hacen que la población de ambas se trate como una sola; además de compartir el sistema de transporte público, miles de personas se movilizan diariamente entre estos municipios, incluyendo estudiantes a sus colegios. En la figura 1.1 se puede observar el área metropolitana.

Figura 1.1: Área metropolitana de Pereira

Debido a la distribución geográfica alargada de la ciudad, algunas heurísticas como la de dos fases podría no adaptarse bien al problema, ya que generalmente tienen éxito en problemas donde el depósito, en este caso colegio, está centrado en el mapa y los puntos están distribuidos alrededor de este. Más importante aún es que la población objetivo son los colegios de tipo privado, los principales de este tipo tienen en su mayoría la característica de encontrarse a las afueras de la ciudad, especialmente en la vía armenia y vía cerritos como se puede observar en la figura 1.2. Los colegios están marcados con estrellas.

Figura 1.2: Mapa de Pereira con principales colegios privados

En la imagen se encuentran los colegios más importantes y los más reconocidos de la ciudad. Vía Cerritos, parte izquierda del mapa, se encuentran el Liceo Campestre

(fuera de la imagen debido a la lejanía), Liceo Merani, Colegio Saint Andrews, Colegio La Salle, Colegio General Rafael Reyes, Colegio Pino Verde y Liceo Inglés. En la vía Armenia, parte inferior derecha, están el Colegio Saint George, Colegio Angloamericano, Liceo Francés, Colegio Abraham Lincoln, Liceo Taller San Miguel y el Colegio Sagrados Corazones. Otros colegios en el mapa son las Bethlemitas, el Calazans, la Enseñanza, el Gimnasio Pereira, las Franciscanas y el Salesiano.
En la ciudad de Pereira por lo regular los colegios privados prestan el servicio de transporte a sus estudiantes; en la tabla de estudiantes del anexo se listan los principales colegios privados de la ciudad y una estimación del número de estudiantes de cada uno, basada en el número de los mismos que presentaron las pruebas Saber 11 en los últimos dos años. Estas pruebas son obligatorias para que los estudiantes de último año de secundaria se puedan graduar y el estado proporciona toda la información sobre cada colegio [22], por lo cual es una buena fuente para hacer la estimación. El cálculo se realizó tomando el promedio de estudiantes de último grado en los últimos 2 años y multiplicándolo por la cantidad de grados que ofrece el colegio, información tomada de [3]. Partiendo de esta estimación se calculó un promedio de 498 estudiantes por colegio, se espera por lo tanto que las soluciones encontradas por el aplicativo sean buenas para problemas de hasta este tamaño. Se tomaron los años 2012 y 2013 ya que no hay suficiente información en la plataforma del ICFES para el 2014.

Capítulo 2

Métodos de resolución

2.1. Introducción

Para la solución de problemas como el VRP se usan algoritmos de inteligencia artificial ya que su rango de solución es demasiado amplio.

2.2. Representación

Se representa el problema de la siguiente manera, se tiene un mapa de Pereira con el colegio úbicado en cuestión, luego se ubican los estudiantes en el punto donde los recogería la ruta. En este caso particular se tiene un VRP donde se encuentran varias rutas con una capacidad específica, que debe ser suficiente para llevar el número de estudiantes ubicados en el mapa. Se desea encontrar la mejor distribución de los estudiantes en cada una de estas rutas para que se consiga la mínima distancia de recorrido total.

2.3. Evaluación

Para evaluar la función objetivo se toma la distancia *Harvesine*, desde el primer estudiante, seguido del segundo la cual se suma con el resto de las distancias siguiendo el orden de estudiantes asignados hasta completar . Y se suman las distancias de cada rutas, otorgando la distancia total recorrida.

2.4. Algoritmos Usados

- Algoritmo Genético debido a su gran cantidad de variables, permite flexibilidad.
- Simulated Annealing (Recocido Simulado), ya que este ha obtenido de los mejores resultados para VRP.

2.5. Mejoras

- Aplicar el algoritmo a nivel de rutas, se puede hacer solo al final o como una mutación.
- Aplicar algoritmos como 2-opt y 3-opt estos son algoritmos de mejora y pueden disminuir el costo de la solución final dada por el algoritmo principal, también pueden ser aplicadas en ciertas iteraciones de los algoritmos, por ejemplo como una mutación en el caso del algoritmo genético.

Capítulo 3

Algoritmo Genético

3.1. Introducción

Los algoritmos genéticos son inspirados en la naturaleza, específicamente en la evolución. Estos simulan el recorrido que han tenido durante millones de años diferentes organismos para adaptarse a su entorno y convertirse en estructuras más complejas, con mejores características que les permitan sobrevivir.
En la naturaleza este proceso se cumple gracias al cruce entre diferentes organismos de la misma especie, a la selección natural y a algunas mutaciones que pueden acelerar el proceso de mejora. El cambio real ocurre a nivel de genética, en donde los cromosomas que tienen la información del organismo se toman de los predecesores, formando un nuevo conjunto que definirá las características del descendiente.
En la computación se tiene la ventaja de poder simular el algoritmo en un ambiente controlado y con organismos de complejidad y cantidad reducida, en este caso de 50 a 1000 cromosomas con información básica. Este proceso de millones de años con las características descritas se puede simular en un tiempo de horas o incluso minutos. Con respecto al ambiente controlado se pueden definir cuantas generaciones simular, el tamaño de la población de la especie, que tipo de selección se aplicará para escoger que instancias de soluciones sobreviven y la forma en que estas se cruzarán para crear las nuevas generaciones; también se pueden definir el tipo de mutaciones y la frecuencia de estas. Estas posibilidades hacen de los algoritmos genéticos una herramienta flexible y poderosa, que bien configurada puede aplicarse a casi cualquier problema, el reto está en escoger las diferentes variantes correctamente y evitar mínimos locales.

3.2. Inicialización de Población

El primer paso para encontrar una solución lo más cercana posible a la óptima es definir una población inicial. Esta consta de un conjunto de n individuos, donde cada individuo es una representación de una posible solución al problema. Como se explica en el capítulo 2, sección 2.2, para el problema actual las soluciones se representan como una lista de paquetes o estudiantes que indica el orden en que estos deben ser recogidos.
Para aplicar el algoritmo a este problema se crea una solución aleatoria para cada individuo de la población, cada una de las soluciones es válida, es decir que contiene a cada uno de los puntos que deben ser asignados a las rutas.

3.3. Selección

El primer punto crucial del algoritmo es la selección. Se debe encontrar un equilibrio entre tener soluciones diferentes entre la población y tener soluciones cada vez mejores. Aquí se tienen algunas variables y puntos a tener en cuenta. Entre las posibles formas de hacer la selección se tienen las siguientes:

1. Por ruleta: A cada individuo.se le asigna una parte proporcional a su aptitud en una ruleta simulada que tiene valores de 0 a 1; luego se genera un número aleatorio en el intervalo y se toma el individuo ubicado en esa posición. Este método se torna bastante ineficiente con poblaciones grandes, ya que a cada uno de los individuos tocaría que agregarles una parte proporcional de la ruleta, por lo que no será tomado en cuenta para el problema actual.

2. Determinística: Donde solo se escoge los mejores individuos de manera elitista.

3. Probabilística: Se toman igualmente un número de individuos al azar pero se selecciona el mejor con probabilidad p, que se tiene normalmente entre 0.5 y 1, favoreciendo al más apto en la mayoría de las ocasiones.

4. Agresiva: Se toman los n mejores o un porcentaje de los mejores de la población.

En los 3 últimos casos casos hay decisiones adicionales que tomar, ¿Cuantos individuos deben competir a la vez?, ¿Con qué probabilidad tomar al mejor?, ¿Que porcentaje o cantidad de individuos tomar? Estas variables pueden afectar drásticamente los resultados del algoritmo y su eficacia varía fácilmente dependiente del problema, por lo cual en lo posible se deben probar todas las opciones con diferentes valores para las variables.
Durante las pruebas se descubrió que bajo algunas configuraciones de selección y cruce se pueden empezar a duplicar soluciones, por lo que se determinó necesario buscar y eliminar soluciones repetidas antes de hacer la selección. Esto se tradujo en una mejora considerable en el valor de las soluciones, este proceso se puede tomar como una preselección y nuevamente hay variantes, en este caso con respecto a cuándo dos soluciones se consideran iguales. La primera opción es considerar dos individuos iguales si su plan de distribución es idéntico, es decir que cada ruta contiene los mismos paquetes y los recoge/entrega en el mismo orden. La segunda consiste en ignorar el orden interno de cada ruta, así si dos soluciones tienen los mismos paquetes en las mismas rutas, estas se consideran iguales aún si cada ruta hace su entrega o recogida en orden diferente .

3.4. Cruce

La reproducción o cruce es el otro punto crítico del algoritmo y también se tienen algunas alternativas, las más utilizadas son el cruce en un punto (conocido como SPX por Single Point Crossover) , el cruce en dos puntos (o DPX por Double Point Crossover) y el uniforme (UPX, Uniform Point Crossover). Estas técnicas aplican solo para representaciones binarias de la solución lo cual no es el caso; sin embargo se encontró una forma de adaptar el cruce uniforme a la representación utilizada. Para

el VRP existen 2 operadores comunes los cuales son: cruce de orden (OX por Order Crossover) y cruce de conjunto de aristas (EAX por Edge Assembly Crossover). Tras el cruce se debe definir también cómo se formará la nueva población.

3.4.1. Cruce uniforme y adaptación

Este cruce se hace generalmente con una máscara binaria del mismo tamaño del cromosoma que representa la solución, donde hay un uno se toma el gen de un padre y donde hay un cero se toma el gen del otro; se tiene la ventaja de que se pueden generar dos descendientes a la vez si se intercambian los papeles de los padres.
El problema es que la representación del problema no es binaria, cada gen debe estar una única vez en la solución y este cruce generaría genes repetidos en una cadena, lo cual no es una solución válida. Para resolverlo se adaptó el algoritmo así:

1. Se crea una cadena binaria aleatoria de la longitud del vector.
2. Donde hayan unos se toma el gen del padre
3. El resto de las posiciones se llenan tomando los genes faltantes en el orden que los tenga el segundo padre.
4. Se hace el mismo proceso invirtiendo los padres.

3.4.2. Cruce de orden

El OX selecciona dos puntos de corte en el cromosoma de los padres, la subcadena de uno es copiada en donde correspondería la del otro y los demás genes se agregan al nuevo individuo de acuerdo al orden en que aparecían en el segundo padre. Para aplicar este método se tomaron los dos puntos de corte aleatoriamente.

3.4.3. Cruce de conjunto de aristas

El EAX es más complejo y fue originalmente diseñado para TSP pero posteriormente adaptado a VRP [35]. En este se combinan las soluciones de ambos padres en un solo grafo, uniendo las aristas de los dos, se crea un conjunto de los ciclos en el grafo seleccionando aleatoriamente nodos de cada padre, luego se escogen algunos de estos ciclos para formar una solución inicialmente incompleta ya que está formada por sub-tours disyuntos, finalmente usando un método codicioso se unen los sub-tours en una solución válida, si esta es mejor que ambos padres será aceptada, de lo contrario el proceso se repetirá hasta que lo sea o se hayan cumplido un número máximo de intentos.

3.4.4. Estrategia destructiva o no destructiva

Luego de realizar el cruce y haber generado un nuevo conjunto de individuos se pueden tomar dos caminos. El primero es agregar los descendientes a la nueva generación aunque estos no sean más aptos que sus padres, esta se conoce como destructiva. Por el otro lado está la estrategia no destructiva en la cual los hijos creados solo se ingresan a la generación si son mejores que sus padres o los los individuos a reemplazar.

3.5. Mutación

Al igual que en la naturaleza se acostumbra a simular una mutación sobre algún individuo, esto se hace con una probabilidad menor a 1 % en lo general. La mutación consiste en alterar algún gen del individuo, para el caso del VRP con la representación que se tomó lo que se puede hacer es cambiar un gen de lugar con otro, esto equivale a cambiar el orden en que se recoge o entrega un estudiante o incluso su ruta con la de otro.
Las mutaciones se hacen con el fin de ampliar el espectro de posibles soluciones a explorar, normalmente una mutación no produce soluciones mejores pero puede hacerlo a largo plazo y ayudar a escapar de mínimos locales.

3.6. Condición de parada

Básicamente existen dos condiciones de parada, la primera es correr el algoritmo hasta que se cumpla un número fijado de generaciones, la segunda opción es parar cuando se haya cumplido una cantidad de generaciones o de tiempo sin que se produzca una mejora.

Capítulo 4

Recocido Simulado

4.1. Introducción

El recocido simulado (Simulated Annealing) es un algoritmo que se basa en una técnica de metalurgia, que a través de un tratamiento térmico puede llevar un material de un estado a otro. El proceso empieza sometiendo el material a una temperatura lo suficientemente alta, así sus átomos alcanzan un alto nivel de energía, permitiendo que estos se reestructuren cambiando su forma inicial. Luego empieza la fase de enfriamiento, donde la energía de estos átomos empieza a decrementar, hasta que llegue a un mínimo estado de energía y el material se cristalice.
Este algoritmo se caracteriza por su manera de escapar de los óptimos locales, esto es logrado gracias a la probabilidad de aceptación de resultados que pueden ser menos buenos que el que se encuentra actualmente. La probabilidad de cambio va variando a medida que la temperatura cambia, si la temperatura se encuentra en altos rangos la probabilidad de que acepte soluciones que no superen o igualen a la actual es mucho mayor que cuando se encuentra a temperaturas bajas.

4.2. Inicialización

El algoritmo requiere unos parámetros de inicio como la temperatura mínima, y una inicial que por lo general empieza con un valor alto con respecto a la temperatura mínima, un número de iteraciones que se van a realizar por cada valor nuevo de la temperatura, un ratio de enfriamiento con el que se va a ir alterando el valor de la temperatura y una solución inicial, que se genera de manera aleatoria, la cual es asignada como mejor solución.

4.3. Alteración de solución

Cuando se empieza el ciclo se escoge algún individuo aleatoriamente y se cambia con otro, se calcula la energía (evalúa la solución) de este nuevo cambio y se compara con la energía actual, si la diferencia de la nueva con la actual es negativa o igual a cero, se toma esta nueva solución y se compara luego con la mejor. Pero si ocurre lo contrario entra la evaluación probabilística (explicada en 4.4), donde hay un chance de que se tome esta solución aunque no sea mejor que la actual. Si esta es mejor se establece como la mejor solución hasta el momento.

4.4. Evaluación probabilística

Para que el algoritmo no se estanque en un mínimo local, se establece una condición de tomar la nueva solución aunque esta no sea más óptima que la anterior:

$$random(0,1) < P \tag{4.1}$$

Donde $random(0,1)$ devuelve un número aleatorio entre 0 y 1, y P es la probabilidad de aceptación la cual es 1 si el valor de la nueva solución es mejor o igual a la actual, en caso contrario devuelve $e^{(\Delta/T)}$ siendo Δ la diferencia entre la solución actual y la nueva, y T es la temperatura actual. Cuando T tiende a cero y la solución nueva es mayor que la solución actual la probabilidad debe tender a cero. Y si la solución actual es mayor la probabilidad tiende a un valor positivo.

4.5. Condición de parada

La primera condición de parada es cuando la temperatura baje hasta alcanzar valor de temperatura mínima, la segunda condición es que la mejor solución no haya cambiado en un determinado número de iteraciones ya que lo más probable es que se haya estancado en un óptimo local y ya no pueda salir de este.

Capítulo 5

Metodología de la investigación

5.1. Diseño metodológico

5.1.1. Hipótesis

Se puede encontrar una buena solución, no necesariamente la óptima, en cuanto a tiempo de respuesta del algoritmo y total de recorrido de las rutas en un VRP utilizando inteligencia artificial.

5.1.2. Tipo de investigación

Este proyecto es cuantitativo ya que la solución propuesta al problema se puede medir en cuanto a la suma de los tiempos promedios de todas las rutas de un colegio antes y después de la implementación del software, adicionalmente si se tienen datos de cuánto tardaría aproximadamente la repartición manual de rutas se puede comparar con el tiempo que tome el software en hacerlo.

5.1.3. Población

Colegios privados de Pereira que presten el servicio de transporte a sus estudiantes.

5.1.4. Unidad de análisis

Base de datos local, generada por el software, donde se tiene información sobre las pruebas realizadas en cada mapa. Mostrando el algoritmo utilizado, el id del mapa, el valor de la función objetivo encontrada, el tiempo de ejecución, la configuración aplicada al algoritmo, número de rutas, capacidad total y porcentaje de ejecución del algoritmo (si se guardó para una prueba determinada).

5.1.5. Muestra

- Colegio Liceo Taller San Miguel, el colegio cuenta un total aproximado de 750 estudiantes, de los cuales 630 pertenecen a la sede campestre ubicada en el kilómetro 8 de la vía a Armenia, y 120 al jardín infantil que se encuentre en el barrio de Álamos.

- Colegio Saint Andrews, ubicado en el kilómetro 7 de la via a Cerritos con un total de 230 estudiantes de los cuales cerca de 180 utilizan el servicio de transporte.

5.1.6. Variables

Las variables que alimentarán el programa para encontrar una solución son las siguientes:

- Número de rutas y capacidad de cada una.
- Número de estudiantes y ubicación de cada uno.
- Ubicación del colegio.

Capítulo 6

Implementación del software

6.1. Análisis

Para el desarrollo del aplicativo se tomaron los requerimientos desde los objetivos del proyecto. Debido a esto no sólo se pensó en el software como tal que diera solución al problema y se vio la necesidad de incluir un módulo de pruebas para correr múltiples ejecuciones con ambos algoritmos y con diferentes configuraciones de manera automática, permitiendo encontrar las mejores combinaciones que se utilizarán a la hora de resolver una instancia específica.
Se vio también la necesidad de un módulo encargado de los mapas ya que se manejan dos tipos, uno básico a modo de matriz que permitirá fácilmente hacer un seguimiento en tiempo real del desempeño del algoritmo cuando se esté en modo de prueba y otro que se basará en los mapas de Google que se utilizará en el modo de producción. Tanto el algoritmo genético como el de recocido simulado interactúan directamente con este módulo por lo que será transparente el tipo de mapa que se esté utilizando.
Los casos de uso se encuentran en las siguientes tablas:

Caso de uso	Crear mapa
Actores	Administrador, Investigador
Referencias	Objetivo 4
Resumen	En este caso de uso se creará una instancia de mapa, ya sea de tipo google maps o por defecto, donde se pondrán los atributos necesarios del mapa y la posición del colegio.
Curso normal de los eventos	
Acción de los actores	**Respuesta del sistema**
1. El actor selecciona el botón “Maps”.	2. El sistema despliega un menú de opciones entre los cuales se encuentra “Create map” y “Manage Maps”.

3. El actor escoge “Create map”.	4. El sistema despliega un formulario para la creación del mapa, con los siguientes atributos: - Name (Nombre del mapa) - Gmaps (Si el mapa es de tipo google maps o default) - Height (Altura, solo requerida si se utiliza sin Gmaps) - Width (Anchura, solo requerida si se utiliza sin Gmaps) - Num of students (Número de estudiantes, solo requerida si se utiliza sin Gmaps) - School x (posición del colegio en x, o la latitud si es de google maps) - School y (posición del colegio en y, o la longitud si es de google maps)
5. El actor llena los campos necesarios y selecciona el botón de “Create it”.	6. El sistema toma los datos ingresados y crea el mapa correspondiente y muestra la sección de “update”.

Cuadro 6.1: Caso de Uso, Creador de mapas

Caso de uso	Manejar mapas
Actores	Administrador, Investigador
Referencias	Objetivo 4
Resumen	En este caso se permite al usuario visualizar los mapas existentes con sus características específicas y realizar algunas acciones con cada uno como visualizar, actualizar y borrar mapa.
Curso normal de los eventos	
Acción de los actores	**Respuesta del sistema**
1. El actor selecciona el botón “Maps”.	2. El sistema despliega un menú de opciones entre los cuales se encuentra “Create map” y “Manage Maps”.

3. El actor escoge “Manage Maps”.	4. El sistema despliega una lista con los mapas existentes, con tres botones al inicio (ver, actualizar y eliminar) y unos atributos específicos de cada mapa: - Name: Nombre del mapa. - Height: Altura, solo requerida si se utiliza sin Gmaps. - Width: Anchura, solo requerida si se utiliza sin Gmaps. - Num of students: Número de estudiantes, solo requerida si se utiliza sin Gmaps. - Best Result: Si se ha corrido algún algoritmo sobre este mapa aparece el mejor resultado obtenido hasta el momento. - Time Best Result: Si se ha corrido algún algoritmo sobre este mapa aparece el tiempo de ejecución del algoritmo que encontró el mejor resultado hasta el momento. - Routes Best Result: Si se ha corrido algún algoritmo sobre este mapa el sistema muestra las rutas en forma de lista, cada elemento (ruta) representando la capacidad de dicha ruta. - Config Best Result: Si se ha corrido algún algoritmo sobre este mapa muestra los valores de algunas variables específicas del algoritmo que encontró la mejor solución.

Cuadro 6.2: Caso de Uso, Manejar mapas

Caso de uso	Ver mapa
Actores	Administrador, Investigador
Referencias	Objetivo 4
Precondición	Haber completado el caso de uso “Manejar mapas”.
Resumen	El usuario podrá visualizar los estudiantes y su ubicación. En caso de que un algoritmo haya corrido se mostrará la mejor solución hasta el momento (Estudiantes se unen con líneas de colores, cada color representando una ruta, de acuerdo a la secuencia encontrada por el algoritmo)

Curso normal de los eventos	
Acción de los actores	**Respuesta del sistema**
1. El actor selecciona el botón ver (que tiene icono de un ojo).	2. El sistema muestra el mapa que contiene el colegio en su dirección específica, y sus estudiantes (si se han establecido) si hay más de un estudiante en un punto, aparecerá un anuncio arriba de este indicando la cantidad de estudiantes presentes. Si se tiene una "mejor solución" el sistema la representará a través de líneas de diferentes colores, cada color representando un bus diferente, que conectan a los respectivos estudiantes de cada ruta en el orden de la solución.

Cuadro 6.3: Caso de Uso, Ver mapa

Caso de uso	Actualizar Mapa
Actores	Administrador, Investigador
Referencias	Objetivo 4
Precondición	Haber completado el caso de uso "Manejar mapas".
Resumen	Se mostrará el mapa con sus estudiantes, mostrando el id de cada uno. Se tendrá la posibilidad de agregar y eliminar estudiantes.
Curso normal de los eventos	
Acción de los actores	**Respuesta del sistema**
1. El actor selecciona el botón actualizar (que tiene icono de un lápiz).	2. El sistema muestra el mapa que contiene sus estudiantes (si se han establecido, con sus respectivos id's).
3. El actor presiona click derecho.	4. Si en el lugar donde se presionó el click existe ya uno o más estudiantes, este será eliminado. Si en caso contrario, no existe ningún estudiante en este lugar, aparecerá un mensaje preguntando el número de estudiantes que se desean agregar en ese punto.
5. El actor proporciona el número de estudiantes a agregar y presiona aceptar.	6. El sistema adiciona un nuevo id en ese punto y agrega la cantidad de estudiantes indicados.

Cuadro 6.4: Caso de Uso, Actualizar Mapa

Caso de uso	Eliminar mapa

Actores	Administrador, Investigador
Referencias	Objetivo 4
Precondición	Haber completado el caso de uso "Manejar mapas".
Resumen	Se eliminará el mapa requerido.
Curso normal de los eventos	
Acción de los actores	**Respuesta del sistema**
1. El actor selecciona el botón eliminar (que tiene una equis de ícono) de un determinado mapa.	2. El sistema elimina el mapa.

Cuadro 6.5: Caso de Uso, Eliminar Mapa

Caso de uso	Definición de costos
Actores	Administrador, Investigador
Referencias	Objetivo 4
Precondición	Haber completado el caso de uso "Manejar mapas".
Resumen	Se permitirá el cambio del costo de ir desde una dirección determinada a otra.
Curso normal de los eventos	
Acción de los actores	**Respuesta del sistema**
1. El actor selecciona el botón cambiar costo (que tiene un ícono de pesos).	2. El sistema muestra un formulario con dos listas conteniendo las direcciones disponibles, un campo "Valor" y un botón "Aceptar".
3. El actor selecciona en la primera lista un origen y en la segunda lista un destino, proporciona el valor del nuevo costo y da click en aceptar.	4. El sistema cambia el valor del nuevo costo y muestra un mensaje de éxito.

Cuadro 6.6: Caso de Uso, Definición de Costos

Caso de uso	Ejecutar múltiples pruebas
Actores	Administrador, Investigador
Referencias	Objetivo 4
Resumen	Se escogerá un mapa a ser probado y se establecerán unas variables para este, se ejecutarán el número de pruebas indicadas y se mostrarán los resultados de estas.
Curso normal de los eventos	
Acción de los actores	**Respuesta del sistema**

1. El actor selecciona el botón “Test”.	2. El sistema despliega un menú que contiene: - Multiple Tests. - Test Results. - Genetic Test. - Simulated Annealing Test.
3. El actor selecciona “Multiple Tests”.	4. El sistema muestra en la parte superior dos botones: - “All results”: Lleva al caso de uso “Resultado de pruebas” - “New Map”: Lleva al caso de uso “Crear Mapa” y luego de estos despliega un formulario con tres campos: - “Routes”: Donde se espera una lista donde cada elemento es un bus y el valor del elemento es la capacidad de este. Ej: “15,15,15” representando tres buses cada uno con capacidad de 15 personas. - “Num Test”: Cantidad de pruebas a correr. - “Mapp”: Se escoge uno de los mapas creados para aplicar las pruebas. Por último mostrará el botón “Test it!” el cual ejecutará las pruebas con los datos del formulario.
5. El actor llena los campos requeridos, escoge el mapa y presiona el botón “Test it!”.	6. El sistema corre las pruebas, y cuando terminan muestra dos tablas: - Datos de Entrada: Contiene los datos ingresados en el formulario, número de estudiantes, número de rutas y capacidad total de los buses. - Resultados: Se encuentran los datos referentes a los algoritmos que ejecutaron las pruebas, en los campos se encuentran: Algoritmo, Promedio (valor promedio del valor objetivo de las pruebas), Tiempo (TIempo promedio en correr las muestras), y por último “Config. Extra” (Las variables específicas del algoritmo).

Cuadro 6.7: Caso de Uso, Ejecutar múltiples pruebas

Caso de uso	Ejecutar Algoritmo Genético
Actores	Administrador, Investigador
Referencias	Objetivo 4

Resumen	Se eliminará el mapa requerido.
Curso normal de los eventos	
Acción de los actores	**Respuesta del sistema**
1. El actor selecciona el botón “Test”.	2. El sistema despliega un menú que contiene: - Multiple Tests. - Test Results. - Genetic Test. - Simulated Annealing Test.
3. El actor selecciona “Genetic Test”.	4. El sistema despliega un formulario con cinco campos campos: - “Mapp”: Se escoge uno de los mapas creados para aplicar las pruebas. - “Routes”: Donde se espera una lista donde cada elemento es un bus y el valor del elemento es la capacidad de este. Ej: “15,15,15” representando tres buses cada uno con capacidad de 15 personas. - “Iterations”: - “Pop Size”: el número de cromosomas que se generarán. - “Elite Factor”: Por último mostrará el botón “Test it!” el cual ejecutará la prueba con los datos del formulario.
5. El actor llena los campos requeridos, escoge el mapa y presiona el botón “Test it!”.	6. El sistema corre la prueba, y cuando terminan muestra la solución y el valor de la función objetivo. Si la solución es mejor que la existente de el mapa, esta será reemplazada.

Cuadro 6.8: Caso de Uso, Ejecutar Algoritmo Genético

Caso de uso	Ejecutar Algoritmo Recocido Simulado
Actores	Administrador, Investigador
Referencias	Objetivo 4
Resumen	Se eliminará el mapa requerido.
Curso normal de los eventos	
Acción de los actores	**Respuesta del sistema**
1. El actor selecciona el botón “Test”.	2. El sistema despliega un menú que contiene: - Multiple Tests. - Test Results. - Genetic Test. - Simulated Annealing Test.

3. El actor selecciona “Simulated Annealing Test”.	4. El sistema despliega un formulario con cinco campos campos: - “Mapp”: Se escoge uno de los mapas creados para aplicar las pruebas. - “Routes”: Donde se espera una lista donde cada elemento es un bus y el valor del elemento es la capacidad de este. Ej: “15,15,15” representando tres buses cada uno con capacidad de 15 personas. - “Iterpertemp”: Número de iteraciones sin cambiar la temperatura. - “Coolrate”: Valor de reducción de la temperatura. - “Inittemp”: Temperatura inicial. Por último mostrará el botón “Test it!” el cual ejecutará la prueba con los datos del formulario.
5. El actor llena los campos requeridos, escoge el mapa y presiona el botón “Test it!”.	6. El sistema corre la prueba, y cuando terminan muestra la solución y el valor de la función objetivo. Si la solución es mejor que la existente de el mapa, esta será reemplazada.

Cuadro 6.9: Caso de Uso, Ejecutar Algoritmo Recocido Simulado

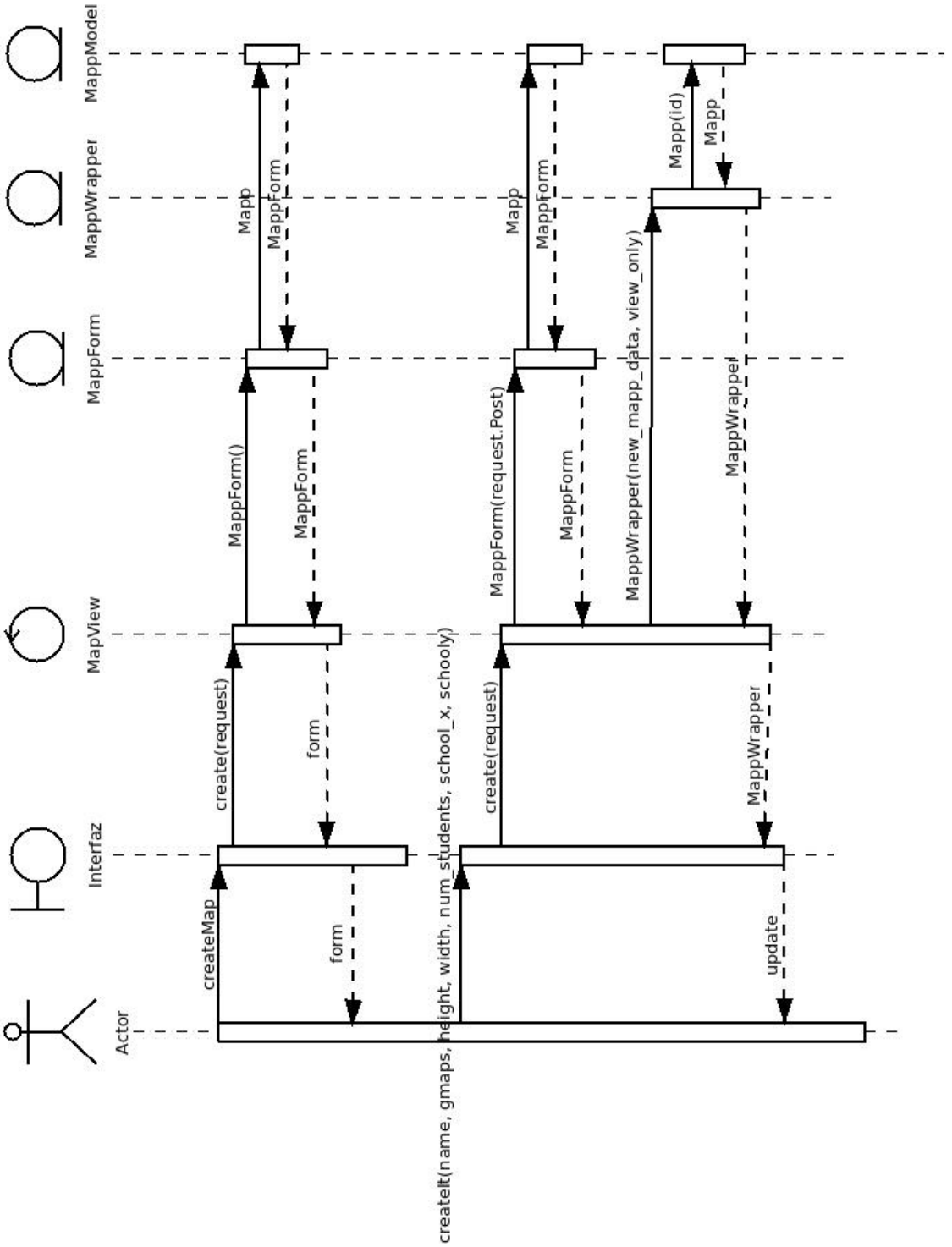

Figura 6.1: Diagrama de Secuencia, Crear Mapa

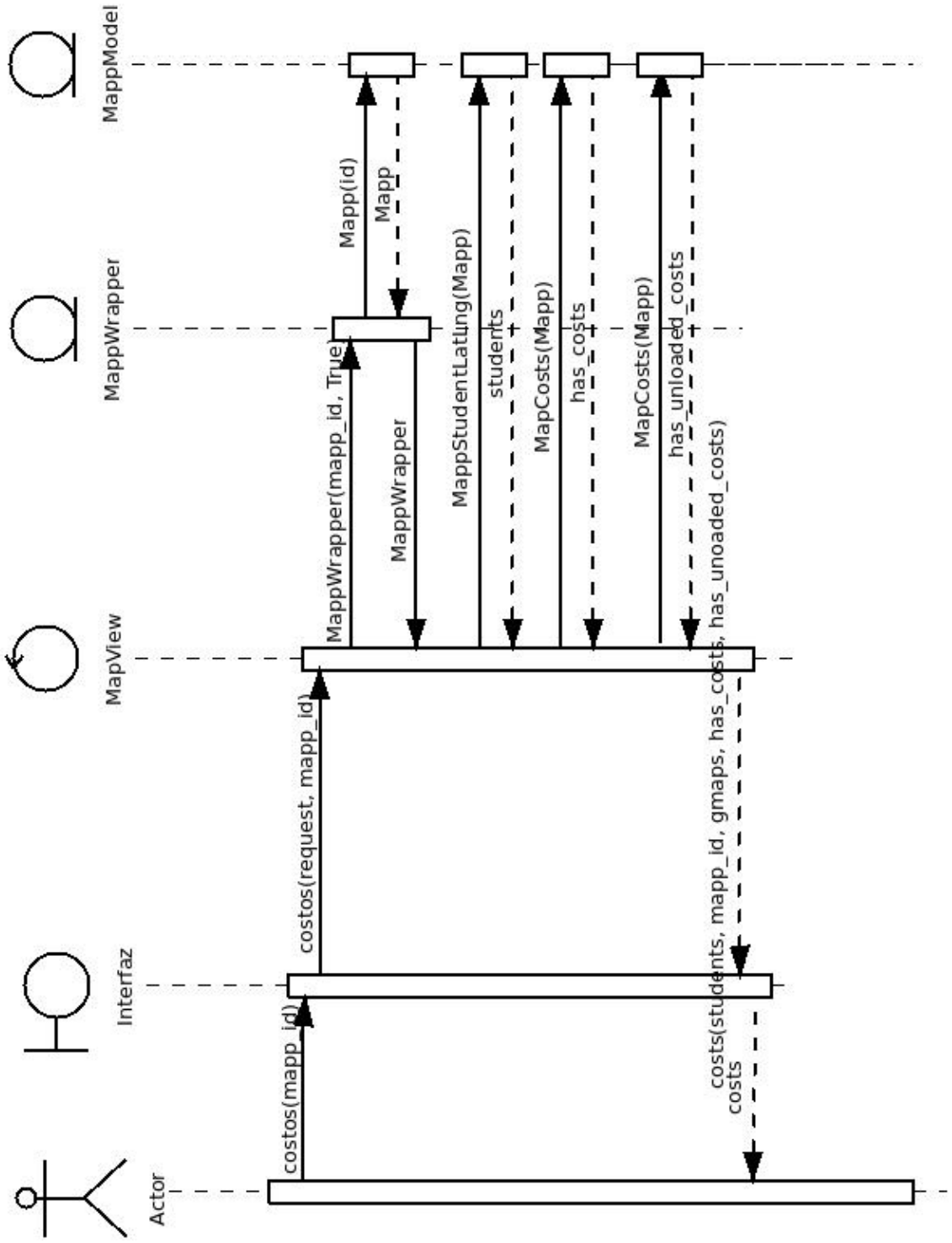

Figura 6.2: Diagrama de Secuencia, Definicion Costos

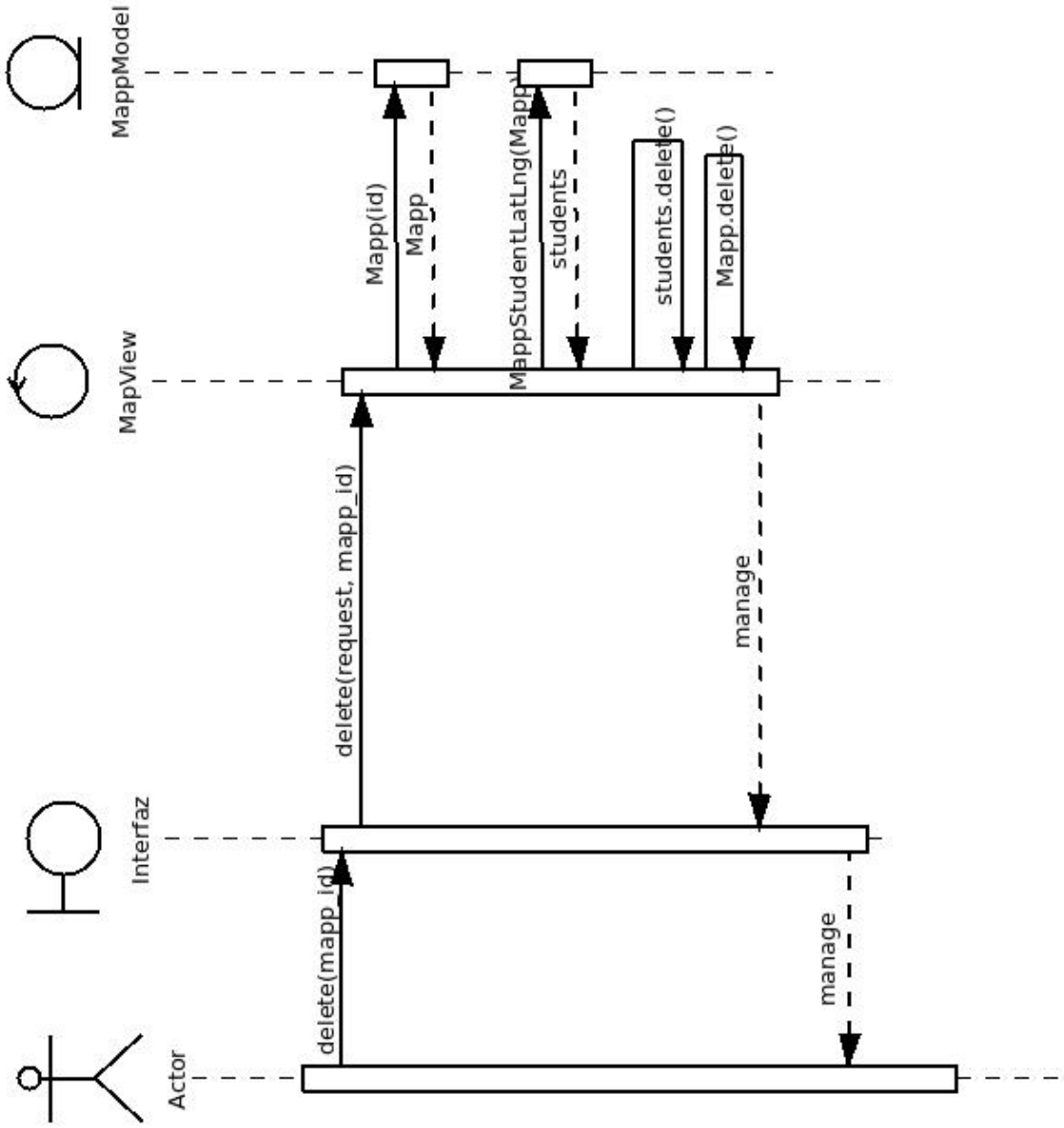

Figura 6.3: Diagrama de Secuencia, Eliminar Mapa

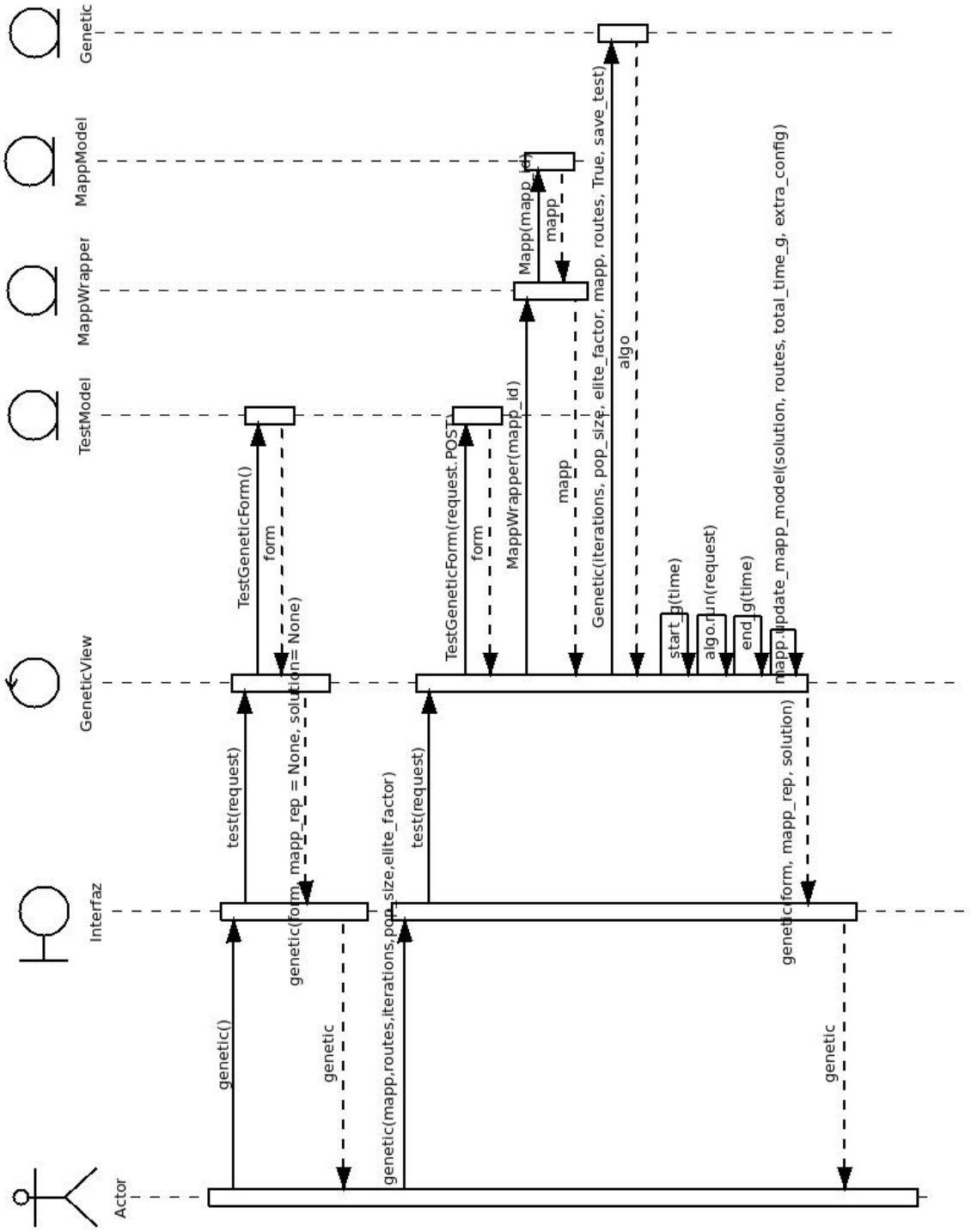

Figura 6.4: Diagrama de Secuencia, Ejecutar Algoritmo Genético

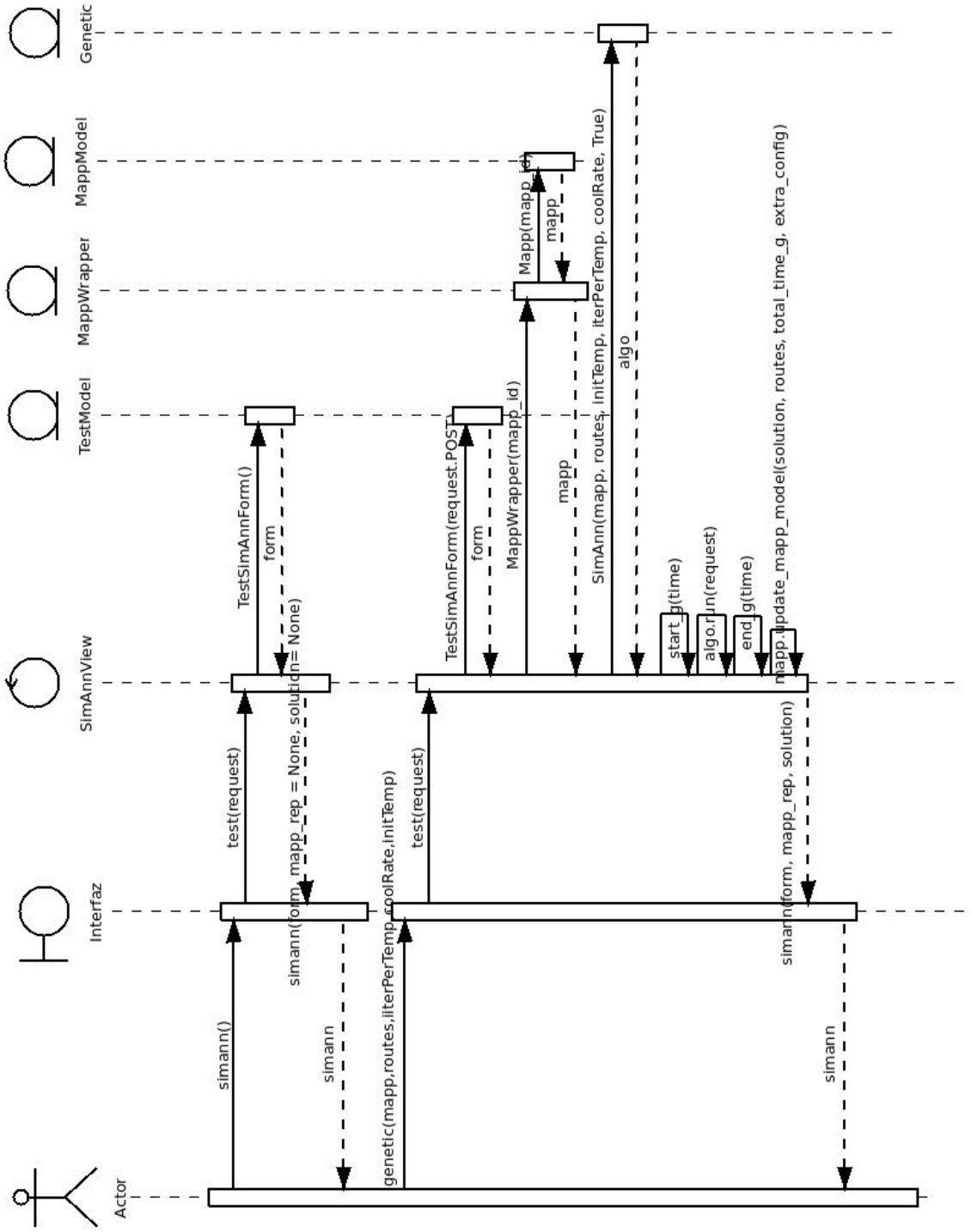

Figura 6.5: Diagrama de Secuencia, Ejecutar Algoritmo Recocido Simulado

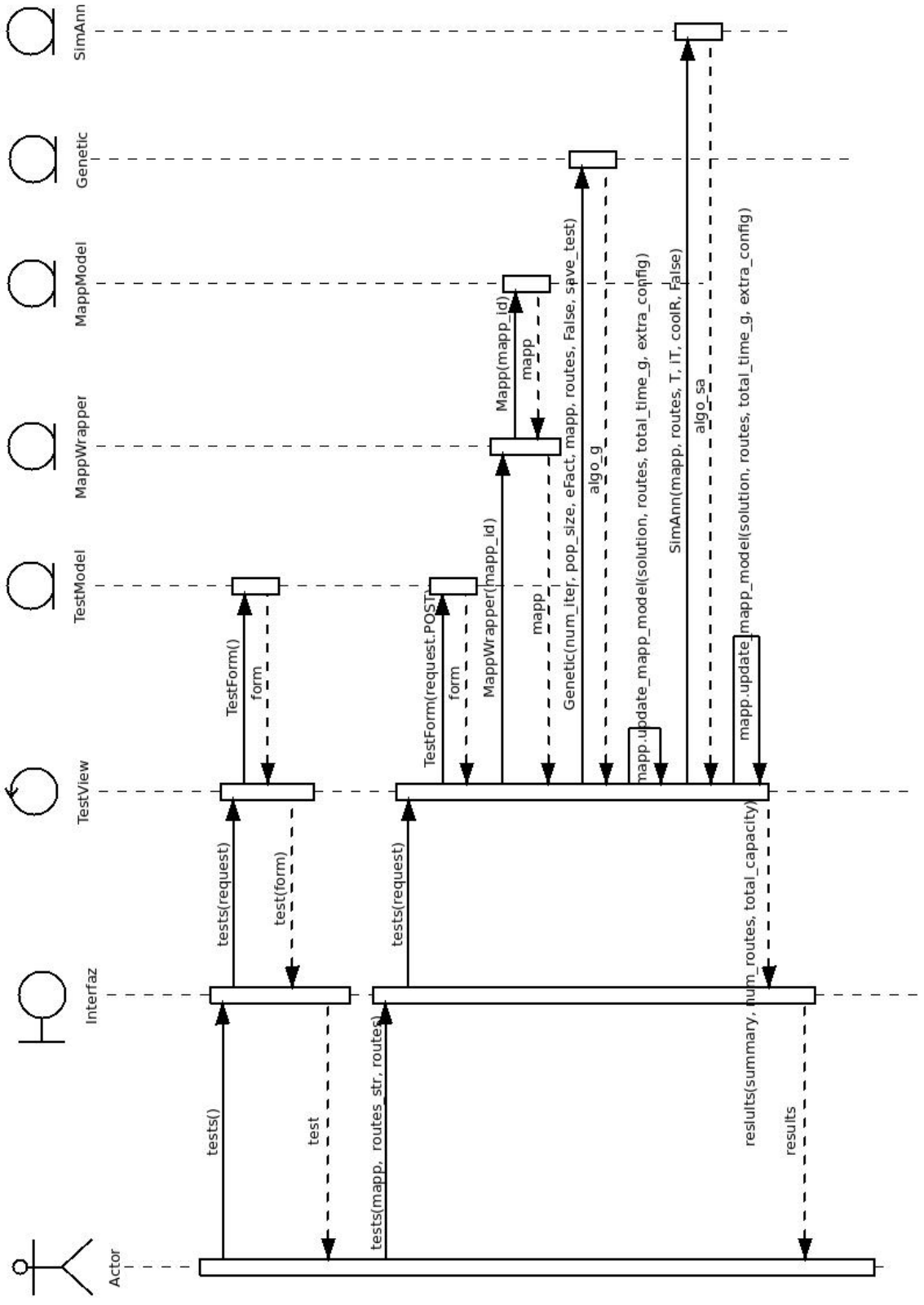

Figura 6.6: Diagrama de Secuencia, Ejecutar Multiples Pruebas

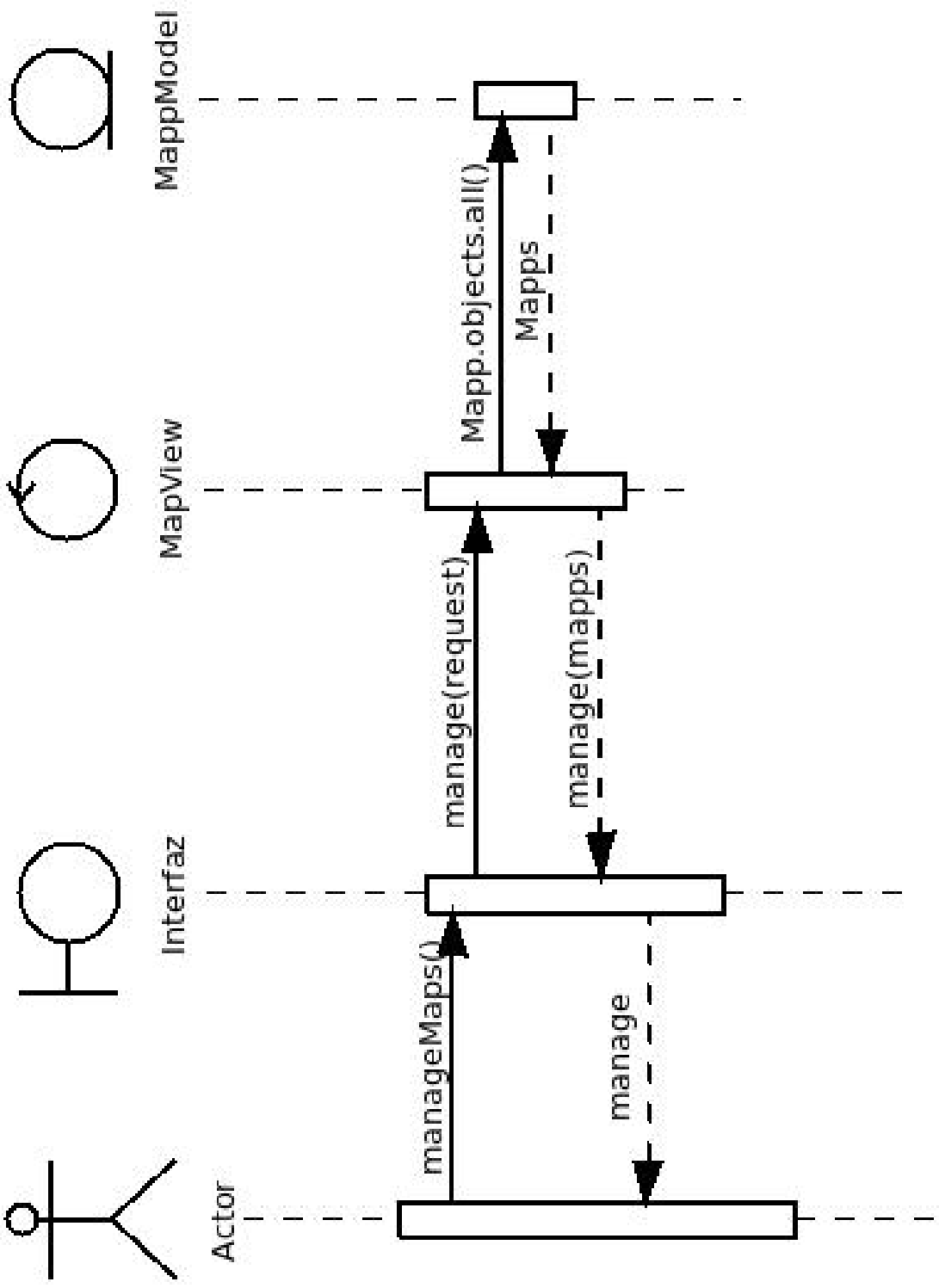

Figura 6.7: Diagrama de Secuencia, Manejar Mapas

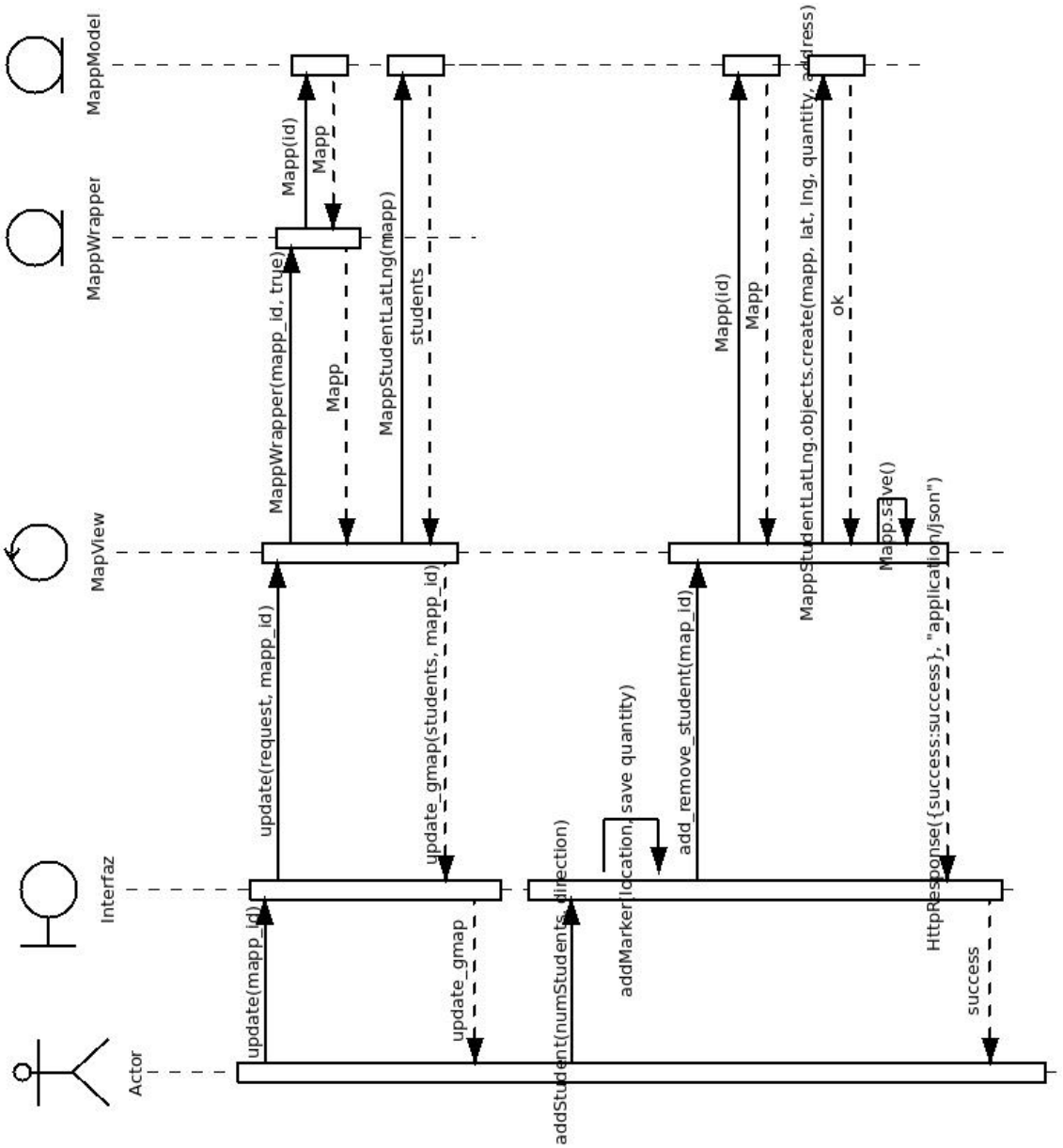

Figura 6.8: Diagrama de Secuencia, Actualizar Mapa

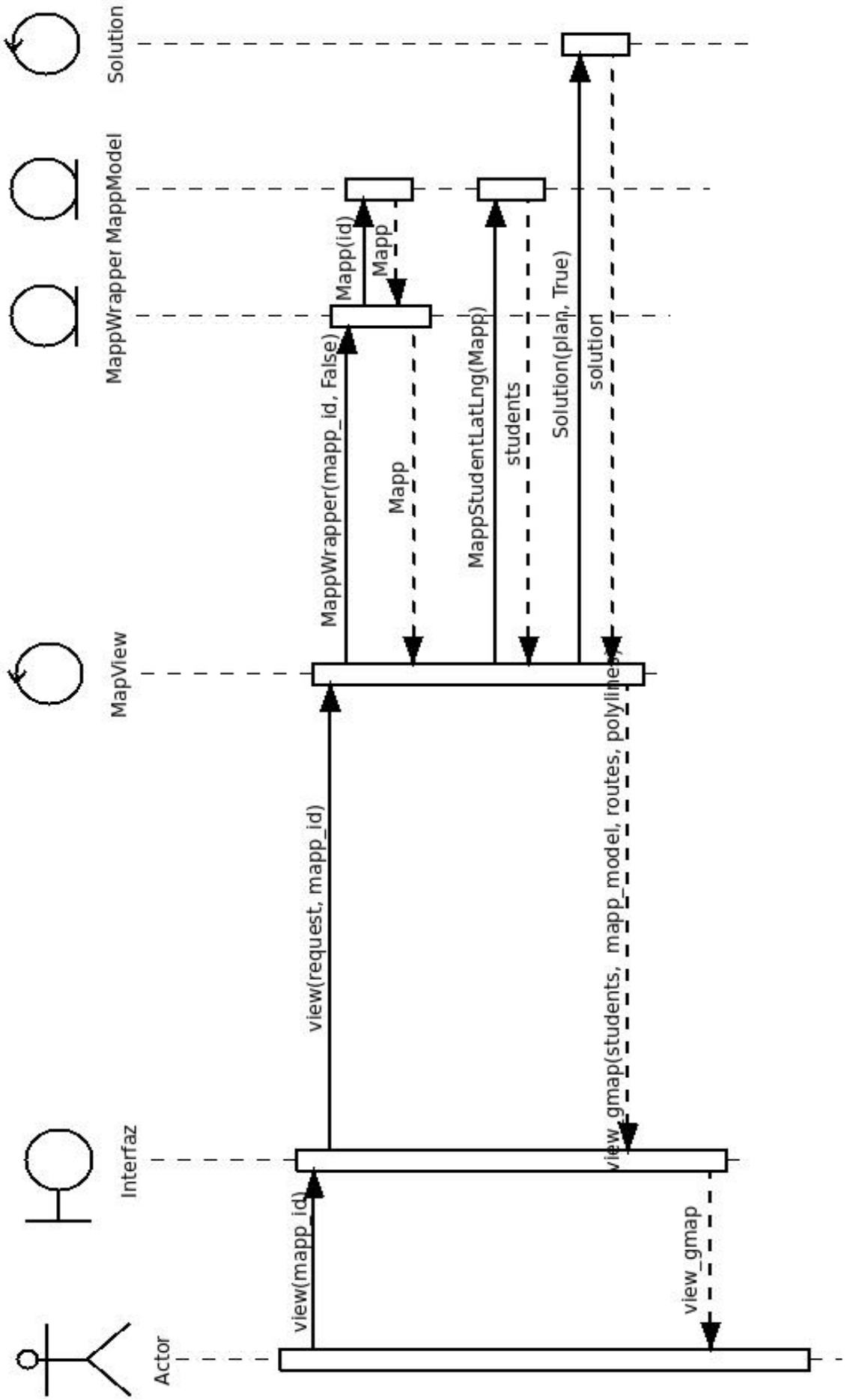

Figura 6.9: Diagrama de Secuencia, Ver Mapa

6.2. Diseño

En el diagrama de clases se puede ver como interactúan estas para complir con los casos de uso y por consecuencia con objetivos del proyecto. Las responsabilidades de cada clase son las siguientes.

Solution

Representación de una posible solución a la distribución de rutas, contiene un arreglo con el plan de distribución y el costo de la misma, cada instancia puede ser creada a partir de un plan o se puede generar aleatoriamente. La clase contiene métodos por lo tanto para crear una solución aleatoria de acuerdo al número de estudiantes y para evaluar su plan según las capacidades de las rutas, el plan se evalúa sumando el costo entre cada estudiante de una ruta y luego sumando los costos de todas las rutas, adicionalmente si para la instancia del problema se definió una ubicación el colegio se sumará el costo del colegio al primer estudiante de la primera ruta, o al último estudiante de la última ruta, según el trayecto.
El costo entre cada ruta se calcula por su distancia dependiendo del tipo de mapa, para los mapas tipo matriz se toma la distancia de manhattan*, para los mapas geográficos se utiliza la distancia harvesiana*.

Student

Esta clase simplemente es una representación de un estudiante, contiene los datos de la posición del mismo en sus coordenadas x,y o latitud,longitud según el tipo de mapa. Además tiene una bandera indicando si es un estudiante fantasma, un identificador único que es asignado por el MappWrapper, un identificador de la familia y el id que le corresponde en la base de datos.
Un estudiante fantasma representa un puesto vacío en un ruta, este tipo de estudiante es necesario ya que la solución debe contener un número de estudiantes igual a la capacidad total de la rutas, los estudiantes de este tipo no son tenidos en cuenta a la hora de evaluar una solución ya que no tiene ningún costo recogerlos, sin embargo los algoritmos los cambian de posición indiferentemente.
El identificador único se utiliza para identificar cada instancia en la solución. El identificador de familia se usa cuando hay más de un estudiante en la misma posición, en la base de datos solo se crea un registro para esos casos pero a la hora de correr el algoritmo se debe crear uno por cada estudiante real, teniendo esto en cuenta la matriz de costos se crea con todas las posibles combinaciones entre familias y no entre estudiantes reales ya que se tendría información duplicada. Este valor por lo tanto se utiliza para acceder a la matriz de costos y los algoritmos lo pueden tener en cuenta a la hora de hacer cambios en una solución, por ejemplo moviendo siempre juntos a estudiantes de la misma familia.
El identificador de la tabla es necesario para un proceso especial en el que se aplica un algoritmo a una sola ruta de la solución .
La clase estudiante se puede inicializar de diferentes formas:

- Enviando las coordenadas
- Sin enviar coordenadas, con lo que generaría una posición aleatoria (Se utiliza para mapas de pruebas en modo de matriz).

- En modo fantasma.

MappWrapper

Esta clase consiste en un capa que se crea en tiempo de ejecución a partir de toda la información de un mapa que se tiene en la base datos y también se encarga de almacenar los datos al crear una nueva instancia.
El MappWrapper contiene una instancia del registro en la base de datos, un diccionario de estudiantes en donde su la llave es el identificador único de cada uno, el número de estudiantes, de familias y de estudiantes fantasmas (puestos vacíos) los cuales son agregados desde cada algoritmo a través de un método que provee esta clase, contiene también una matriz de costos de (n+1)x(n+1) siendo n el número de familias, y la escuela como una instancia de Student.
La matriz de costos indica el costo entre cada par de estudiantes, y en la posición i,i se tiene el costo del estudiante i al colegio. Esta matriz se carga al instanciar un objeto de MappWrapper, si no se tiene la información de los costos esta matriz se cargará inicialmente con valor -1 en todas las posiciones. Cada vez que se crea una nueva solución esta se debe evaluar y allí es donde realmente se utilizan estos valores, en el caso en que se encuentre un -1 se calculará la distancia según el tipo de mapa y se guardará en la matriz. Finalmente cuando un algoritmo termine debe pedirle a su instancia de mapa que guarde los valores de la matriz a través de un método, este actualizará en la base de datos los valores que no se hubieran calculado antes.

Genetic y SimAnn

Estas clases son las encargadas de correr el algoritmo genético y el de recocido simulado respectivamente, ambos implementan un método constructor que recibe una instancia de MappWrapper, un vector de con las capacidades rutas y los parámetros adicionales de cada algoritmo.
También implementan un método run() que se encarga de correr el algoritmo con los datos del mapa, las rutas y los parámetros dados, este método en ambos casos retorna una instancia de Solution, con la mejor solución encontrada.

6.3. Diagrama de entidad relación

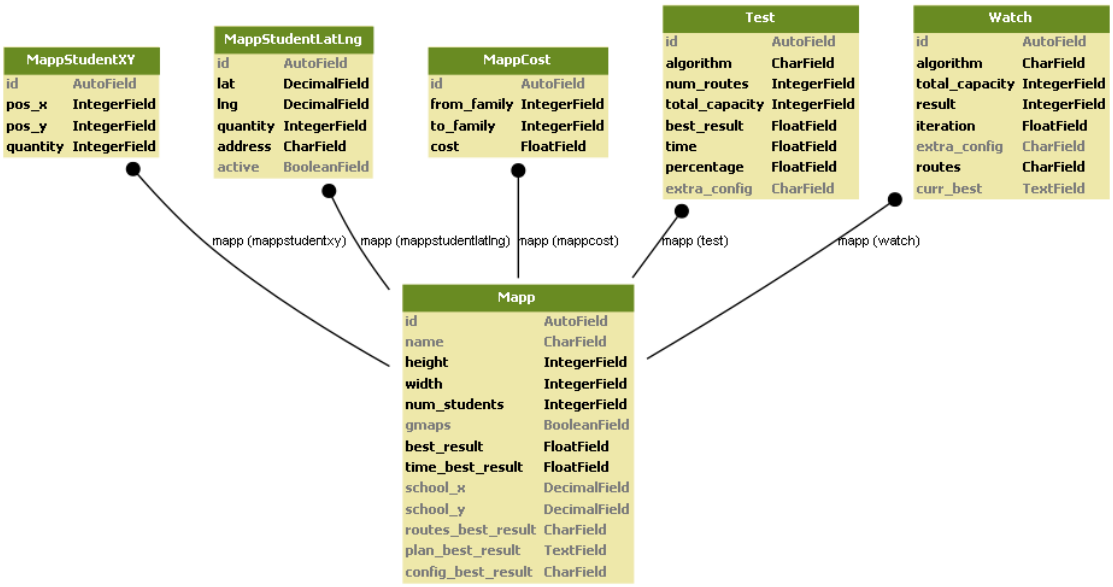

Figura 6.10: Diagrama Entidad Relación

6.4. Pruebas

Para poder llevar a cabo pruebas con datos reales el colegio Liceo Taller San Miguel facilitó los listados de direcciones de sus estudiantes (sólo la direccion). Este colegio tiene una jornada para los estudiantes de primaria y otra para los de bachiller, además de una sede jardín en álamos. Adicionalmente contamos con la colaboración del colegio Saint Andrews. Gracias a esto se tiene la información de 5 conjuntos diferentes de estudiantes que deben ser recogidos: los estudiantes de primaria, los estudiantes de bachillerato y la jornada de la mañana y de la tarde del jardín de Álamos, las cuales tienen muchas diferencias.
En total se tiene entonces 5 casos reales de prueba con las siguientes características:

Descripción	Número de rutas	Número de estudiantes
LTSM Bachillerato sede campestre	19	271
LTSM Primaria sede campestre	24	367
LTSM Jardín Infantil, sede álamos. Jornada mañana	5	44
LTSM Jardín Infantil, sede álamos. Jornada tarde	3	27
Saint Andrews sede campestre	12	158

Cuadro 6.10: Características casos reales

Una ventaja destacable de los casos de prueba con los que contamos es que tenemos un colegio en la vía cerritos y uno vía armenia, donde se encuentra gran parte de la población definida para el proyecto.

Capítulo 7

Análisis de resultados y conclusiones

7.1. Resultados con recocido simulado

El algoritmo de recocido simulado (*Simulated Annealing*) dio resultados muy buenos en tiempos cortos. Aún así, para lograr los mejores resultados se observó que fue con un enfriamiento muy lento como se puede observar en el cuadro 7.1, e iniciando con una temperatura baja (menor que 5). Cuando se realizaron pruebas con temperaturas altas (mayores que 100) se observó que se demora más para poder encontrar buenas soluciones. También se realizaron unas pruebas donde se ejecutaba el algoritmo con porcentaje de enfriamiento alto(0.001), lo cual permite que en muy poco tiempo se baje a una solución buena, y esta solución era pasada a otra ejecución del algoritmo con porcentaje de enfriamiento bajo (0.00001); Se vio que de esta forma no es tan eficiente, ya que se queda enfrascado en un óptimo local más fácilmente. En la figura A.3 se puede ver como con un enfriamiento muy lento, el algoritmo se queda en etapas muy tempranas en un óptimo local (desde el 40 %).
En la tabla A.3 se puede observar la ejecución con una configuración específica, viendo como va cambiando la función objetivo en el porcentaje de ejecución. Este se para en el 57 % porque no cambió la solución por más de $1 * 10^7$ ciclos.
En la figura 7.1 se puede observar que los mejores resultados se dieron al tener una proporción de enfriamiento muy lenta "$1e - 05$", aunque su tiempo de ejecución crece exponencialmente a medida que la proporción de enfriamiento es menor. Así como se observa en la figura 7.2 el mejor resultado se dio al tener el *CoolingRate* menor, y su tiempo de ejecución es el que más tardó.

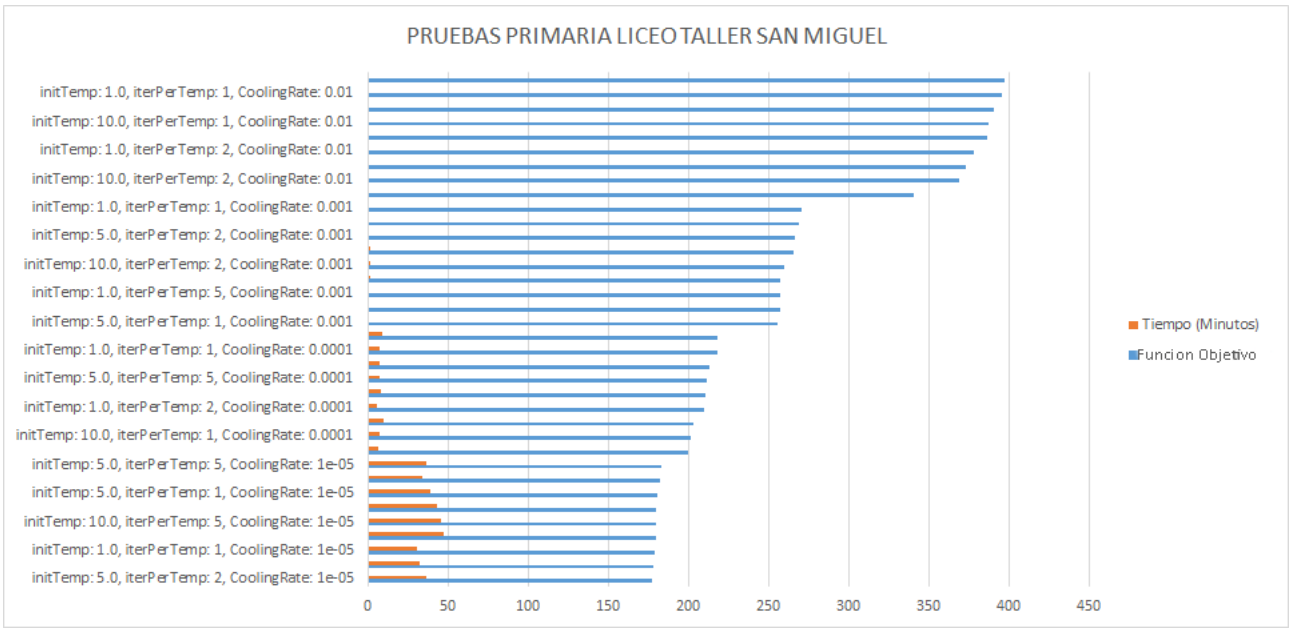

Figura 7.1: Gráfico de Resultados Recocido Simulado, Liceo Taller San Miguel, Primaria

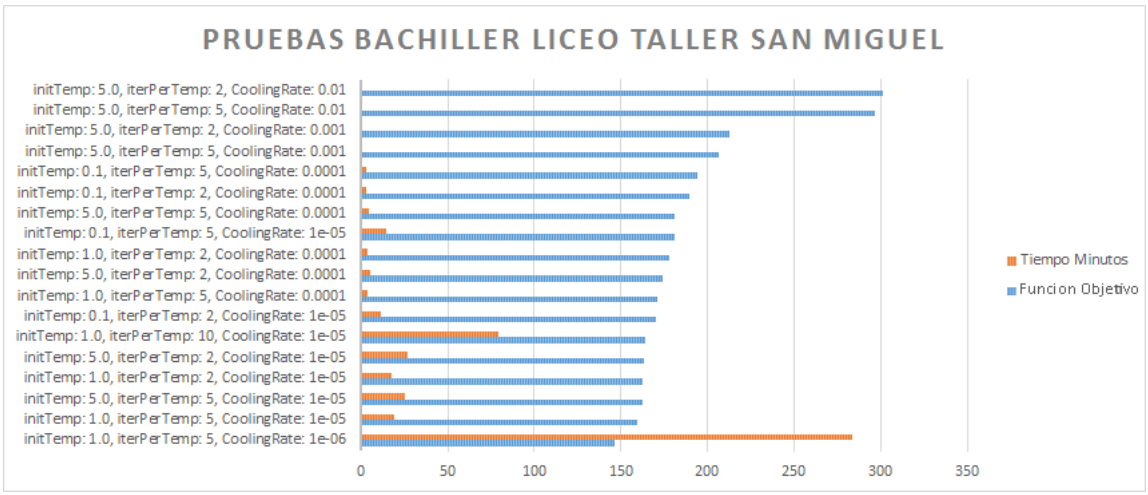

Figura 7.2: Gráfico de Resultados Recocido Simulado, Liceo Taller San Miguel, Bachillerato

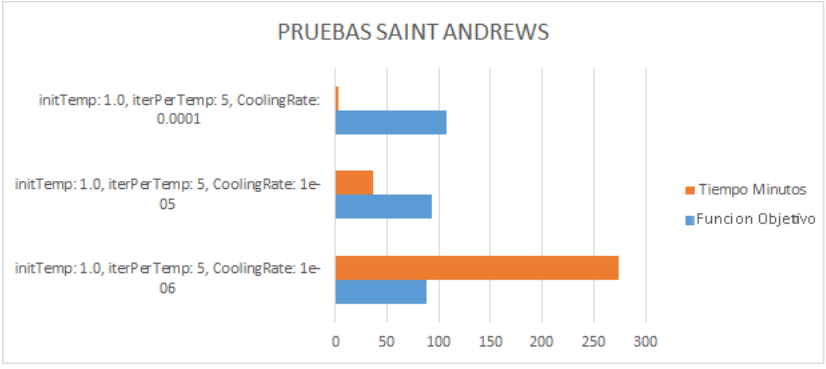

Figura 7.3: Gráfico de Resultados Recocido Simulado, Saint Andrews

7.2. Resultados con algoritmo genético

Para realizar las pruebas con el algoritmo genético se definieron algunas de las variables de acuerdo a lo que se definió para el desarrollo y lo que se observó durante esta etapa. Para la selección optamos por la estrategia determinista, lo que introdujo unas de las variables que nombramos *Elite Factor*, indicando el porcentaje de población considerada como élite que formará la próxima generación. El cruce se hizo con el algoritmo OX, o cruce de orden.
Como mutación se implementó el algoritmo k-opt y se aplicaba a individuos al azar, sin embargo los resultados se veían desmejorados debido a que la solución caía más fácil en mínimos locales, en la mayoría de las ocasiones aplicar el k-opt producía una mejora al aplicarse sobre el mejor resultado encontrado por el algoritmo genético, sin embargo esta estrategia se cambió y en su lugar se definió correr el algoritmo de recocido simulado sobre la mejor solución del genético, esta estrategia tuvo los mejores resultados.
Sobre el tamaño de la población y el número de generaciones se decidieron valores altos que no hicieran que el tiempo de ejecución sobrepasara las 6 horas. Estos fueron un tamaño de 1000 individuos por 5000 generaciones.
Otra variante con la que se experimentó fue hacer un ajuste en el cruce de acuerdo a qué tan avanzadas estaban las generaciones. El cruce OX toma dos puntos de uno de los padres aleatoriamente para tomar una sub-cadena que será parte del nuevo individuo, el ajuste consiste en que a medida que las generaciones avancen esta cadena debe ser mas pequeña, esto se logró multiplicando la longitud de la cadena formada al azar por el porcentaje de generaciones restantes y reduciéndola a este nuevo tamaño.
Fueron definidas entonces las siguientes estrategias:

- Selección determinista
- Cruce OX
- Aplicar algoritmo simulado sobre la mejor solución encontrada

Constantes:

- Población: 1000
- Generaciones: 5000

Variables:

- Factor de elite (EF) = [0.1, 0.15, 0.2, 0.25, 0.3]
- Ajuste porcentual en cruce (A %) = [Si, No]

Con estas configuraciones se probaron los mapas de bachillerato del Liceo Taller San Miguel con 267 estudiantes y del Colegio Saint Andrews con 158.
Para el LTSM se corrió 12 veces el algoritmo con cada una de las configuraciones, los resultados mostraron que al final de la ejecución del algoritmo genético los mejores resultados se tenían usando un factor elite de 0.3, y que en 4 de las 5 opciones definidas para el factor elite se obtuvo en mejor resultado en promedio usando ajuste porcentual. Por estas razones se hicieron 12 pruebas utilizando el factor 0.35 y 0.4,

en ambos casos con ajuste porcentual. En la figura 7.4 se observa el valor promedio para cada una de las configuraciones según el porcentaje de generaciones, allí se puede identificar fácilmente que la configuración que obtuvo mejores resultados fue con un factor elite de 0.35 y usando ajuste porcentual.
En la figura 7.5 se observa como a medida que se aumenta el factor de elite los resultados mejoran teniendo su mínimo en 0.35, incluyendo no solo en el promedio sino también en el mejor y el peor resultado. Sin embargo los valores al finalizar el algoritmo genético están lejos de los mejores encontrados, es al aplicar el algoritmo de recocido simulado sobre el mejor resultado del genético que se encuentran soluciones realmente buenas y esto según los resultados no depende mucho de la configuración del genético, se puede evidenciar en la figura 7.6, los datos de ambas figuras están en la tabla A.7. En las tablas A.4 y A.5 están los datos de los promedios por porcentaje y en la última fila está el cálculo de la mejora del resultado tras aplicar el recocido simulado, el valor total de recorrido se reduce hasta entre un 35 % y 38 %.
Con respecto a los tiempos cada prueba tardó entre 2.5 y 4 horas en una computadora con procesador i3 con procesadores de 2.5Ghz, usando el 25 % del procesamiento. Las pruebas más demoradas pueden ser ejecutadas en un tiempo menor a una hora en una computadora con mejores características, y podrían ser aún mas rápidas utilizando hilos, técnica que se puede adaptar con facilidad debido a la naturaleza del algoritmo que debe procesar cientos de individuos. Los datos completos de los tiempos se encuentran en la tabla A.6 y están graficados en la figura 7.7. En la gráfica se puede apreciar que el proceso mas demorado es la reproducción y que este tiempo al igual que el total disminuye a medida que se aumenta el factor de elite, y es siempre menor si se aplica el ajuste porcentual.
Para el mapa del colegio Saint Andrews se realizaron 10 pruebas con cada una de las configuraciones, los mejores resultados al finalizar el genético se obtuvieron con factor de elite de 0.2 y 0.30, ambos con ajuste porcentual, este comportamiento se puede ver en las figuras 7.8 y 7.9.
Al igual que en las pruebas del LTSM se encontró que tras aplicar el recocido simulado al mejor resultado del genético, los resultados no dependen de la configuración inicial, figura 7.6. La distancia total se reduce hasta un 50 %, ver tabla A.8.
Los tiempos de ejecución del algoritmo para el mapa del Saint Andrews ocilan entre 80 y 110 minutos, y presentan el mismo comportamiento de ser mas cortos a medida que se aumenta el factor de elite.

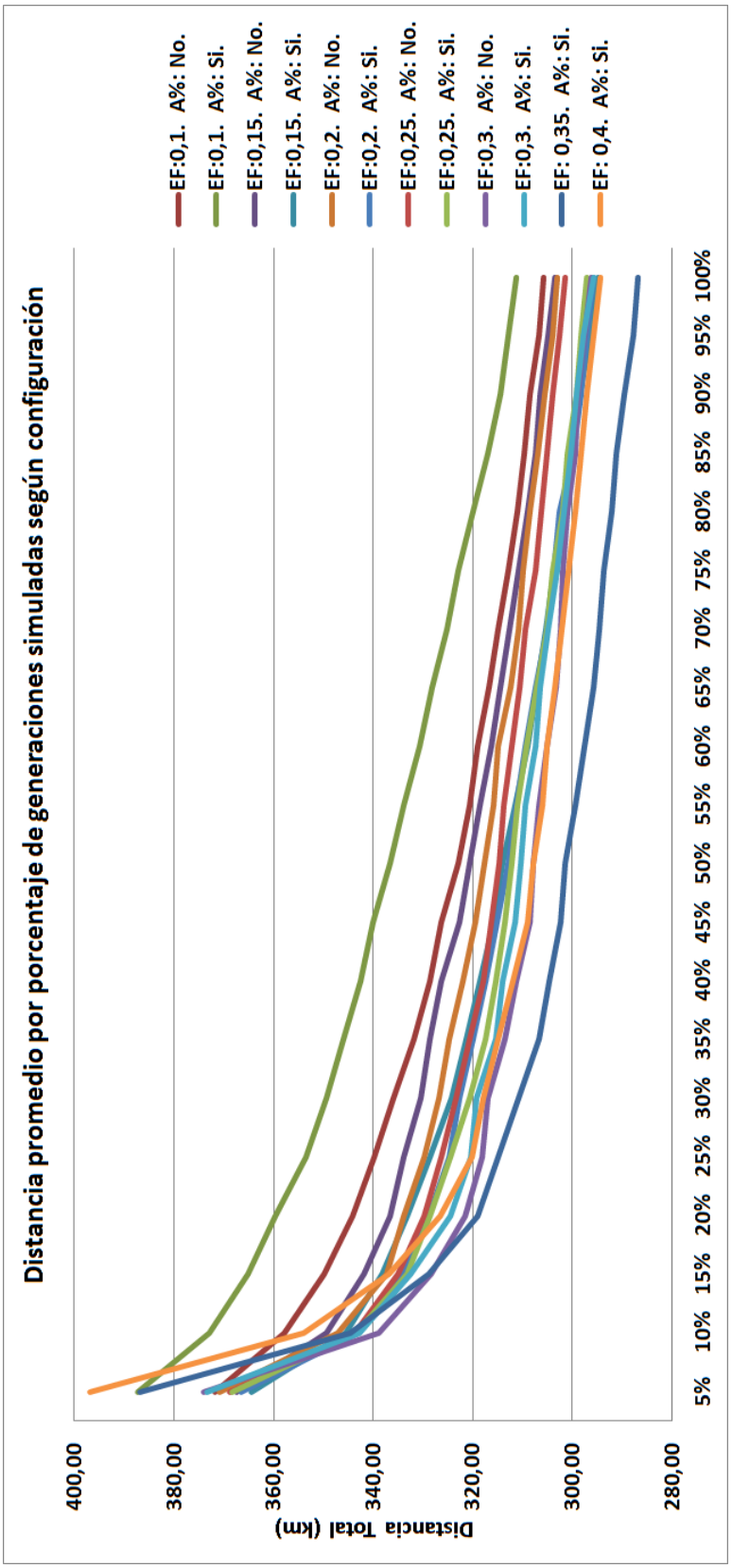

Figura 7.4: Distancia promedio por porcentaje de generaciones simuladas según configuración. Liceo Taller San Miguel, Bachillerato

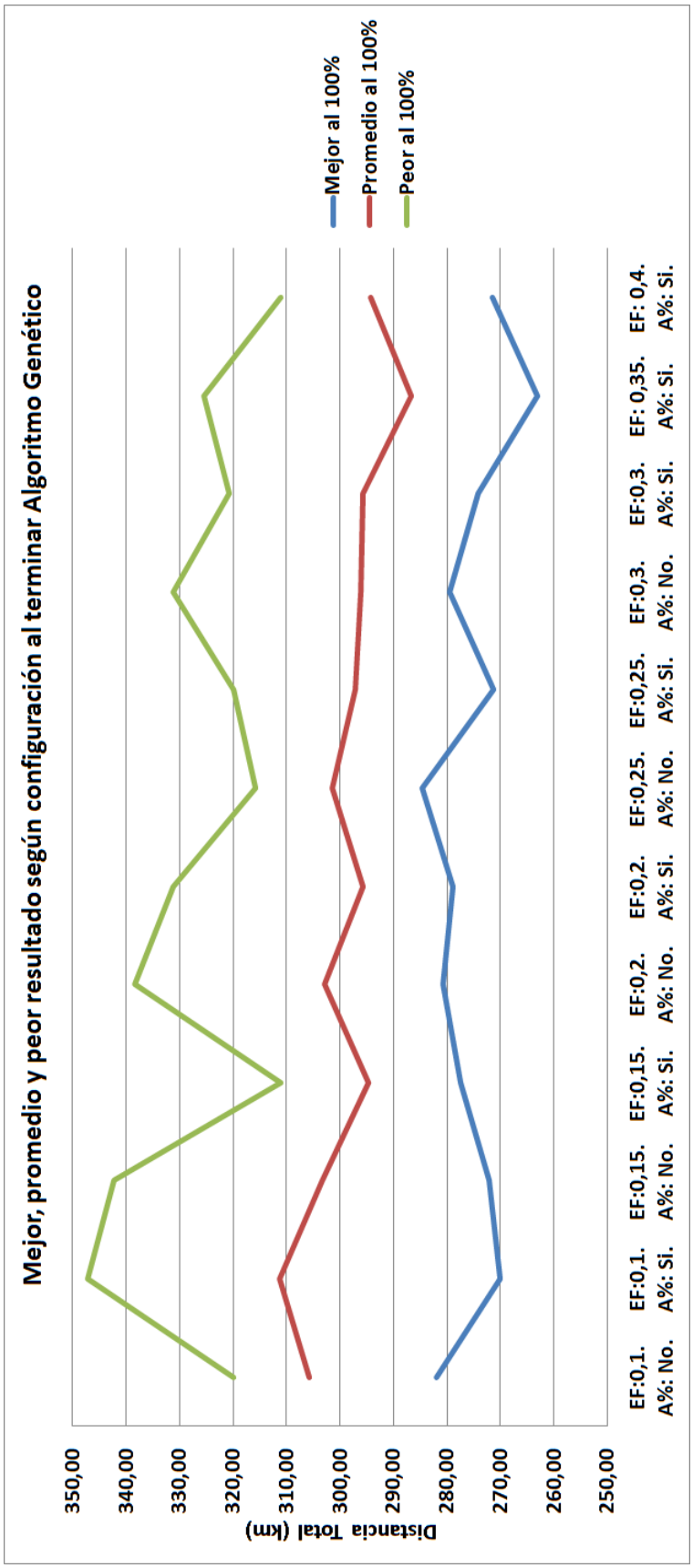

Figura 7.5: Mejor, promedio y peor resultado según configuración al terminar Algoritmo Genético. Liceo Taller San Miguel, Bachillerato

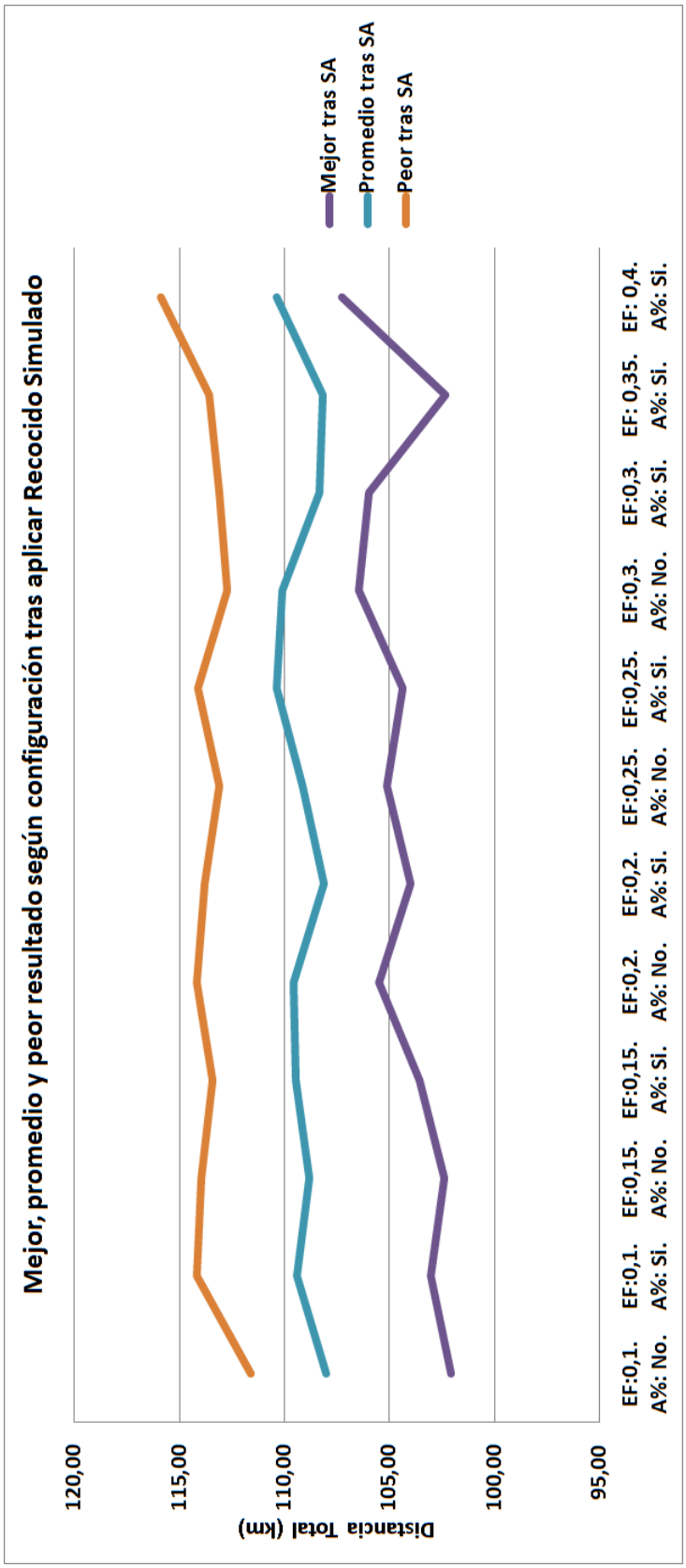

Figura 7.6: Mejor, promedio y peor resultado según configuración tras aplicar Recocido Simulado. Liceo Taller San Miguel, Bachillerato

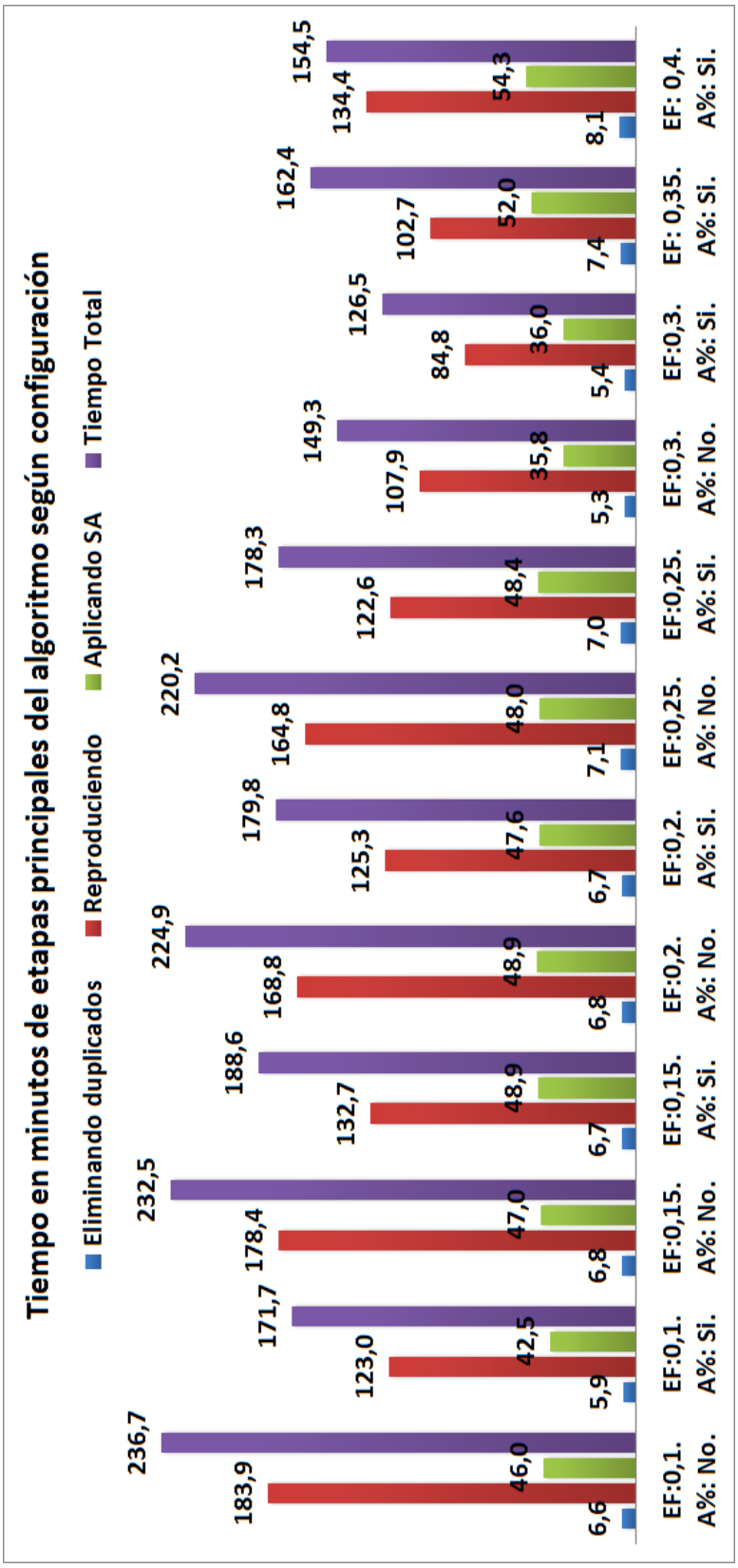

Figura 7.7: Tiempo en minutos de etapas principales del algoritmo según configuración. Liceo Taller San Miguel, Bachillerato

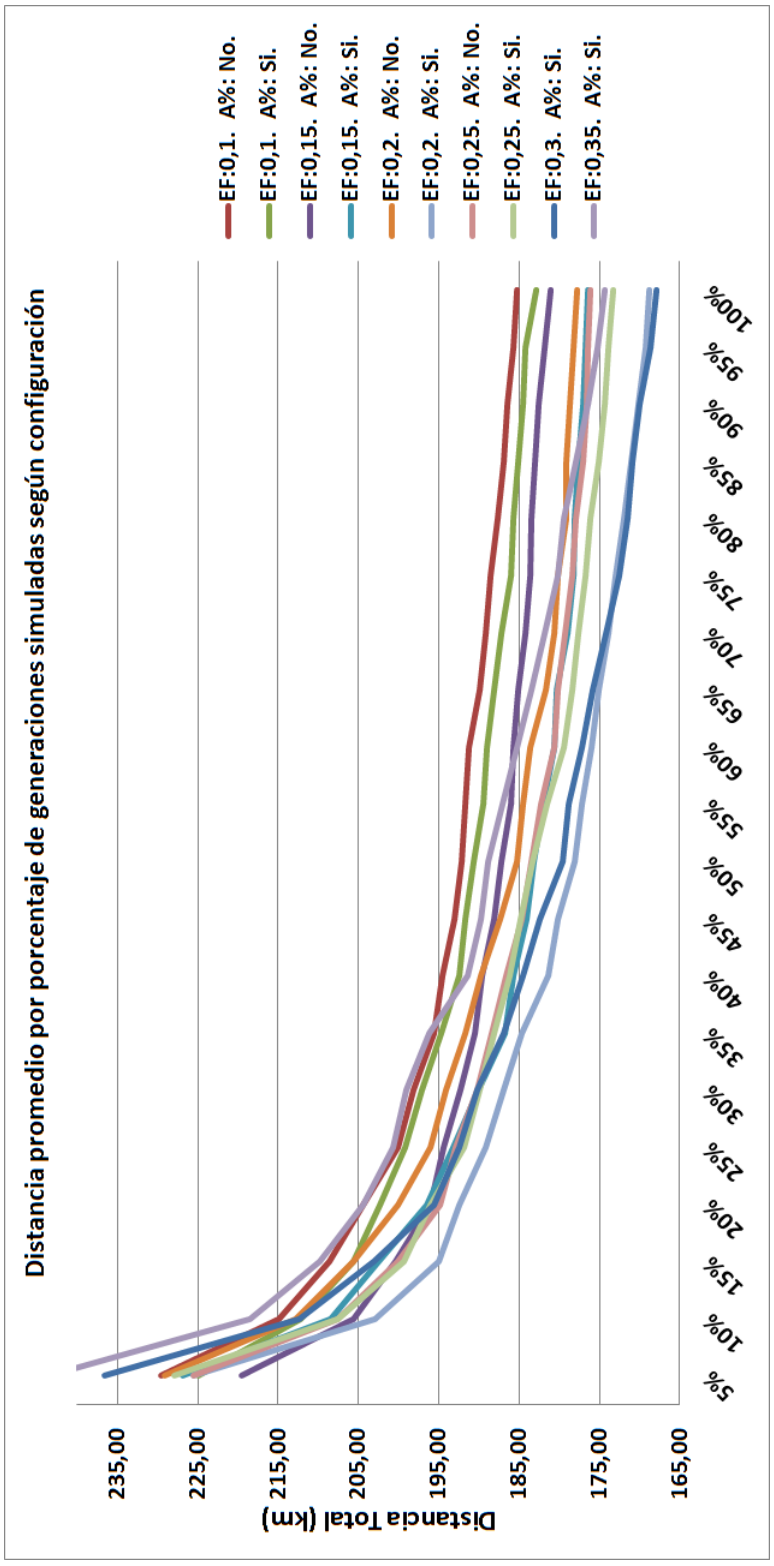

Figura 7.8: Distancia promedio por porcentaje de generaciones simuladas según configuración. Colegio Saint Andrews

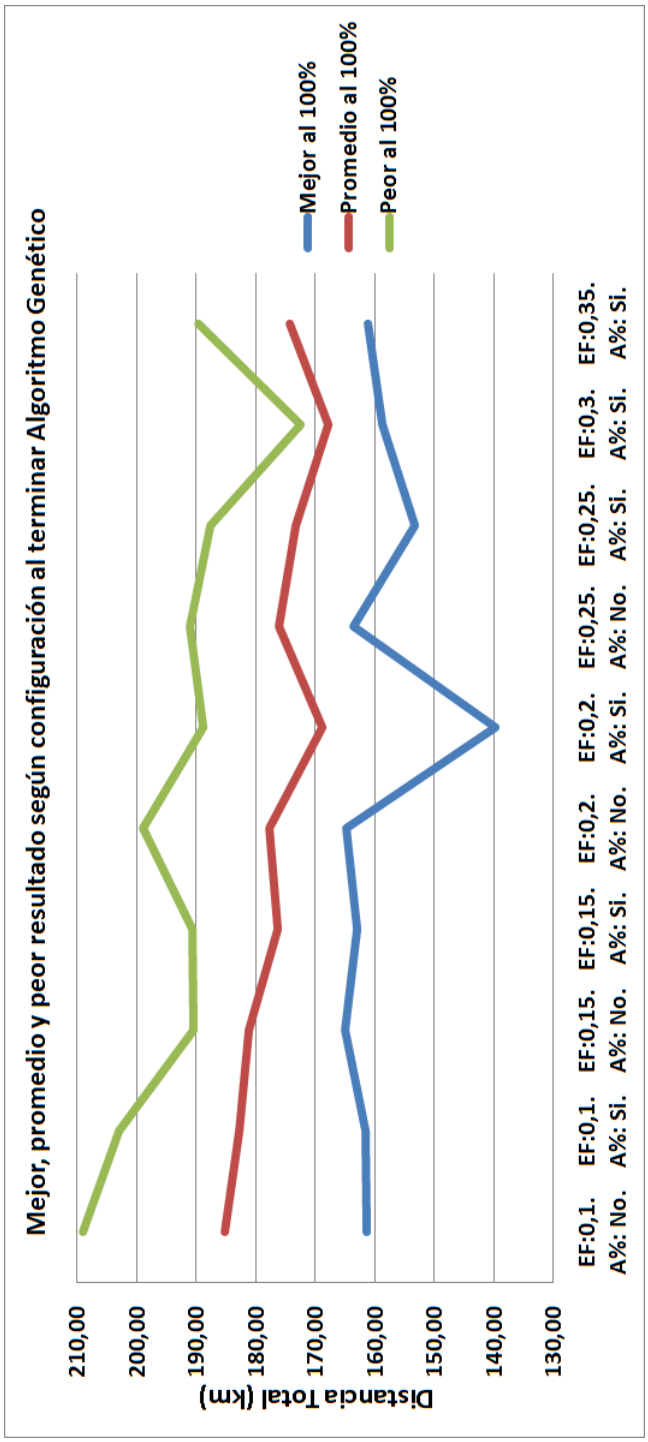

Figura 7.9: Mejor, promedio y peor resultado según configuración al terminar Algoritmo Genético. Colegio Saint Andrews

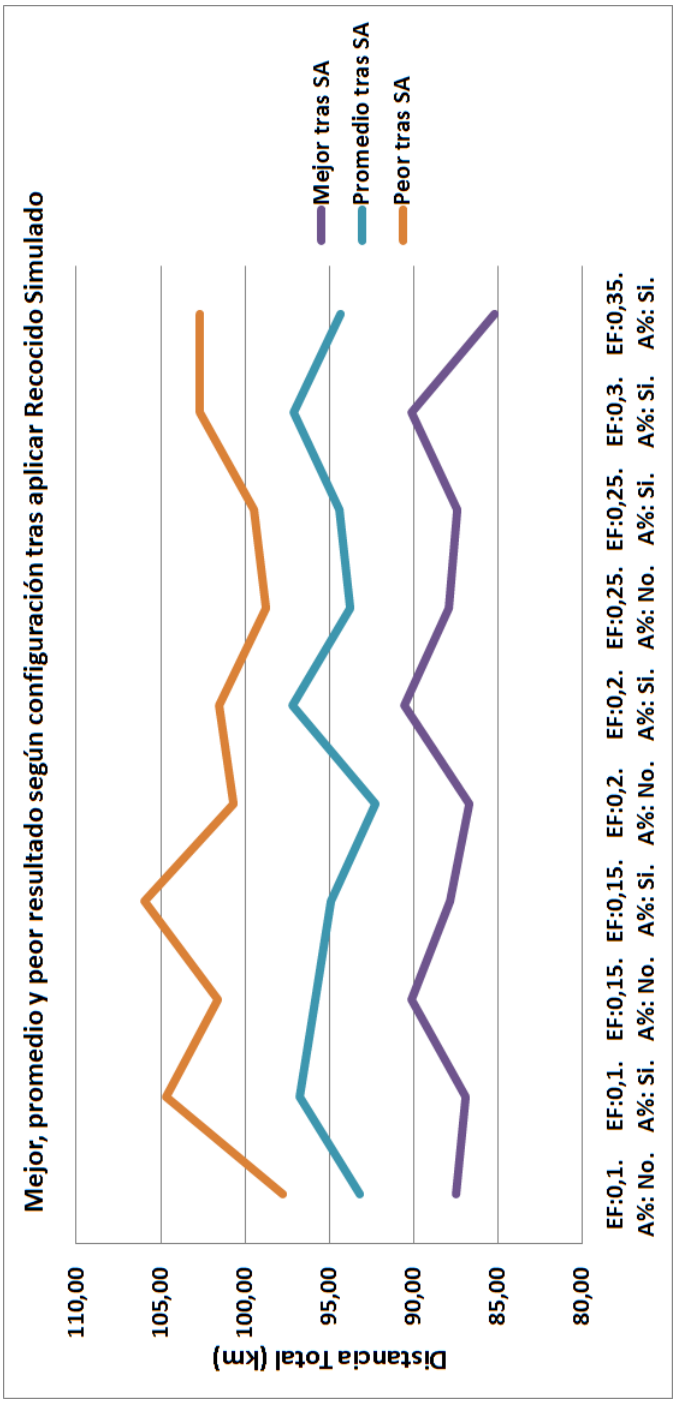

Figura 7.10: Mejor, promedio y peor resultado según configuración tras aplicar Recocido Simulado. Colegio Saint Andrews

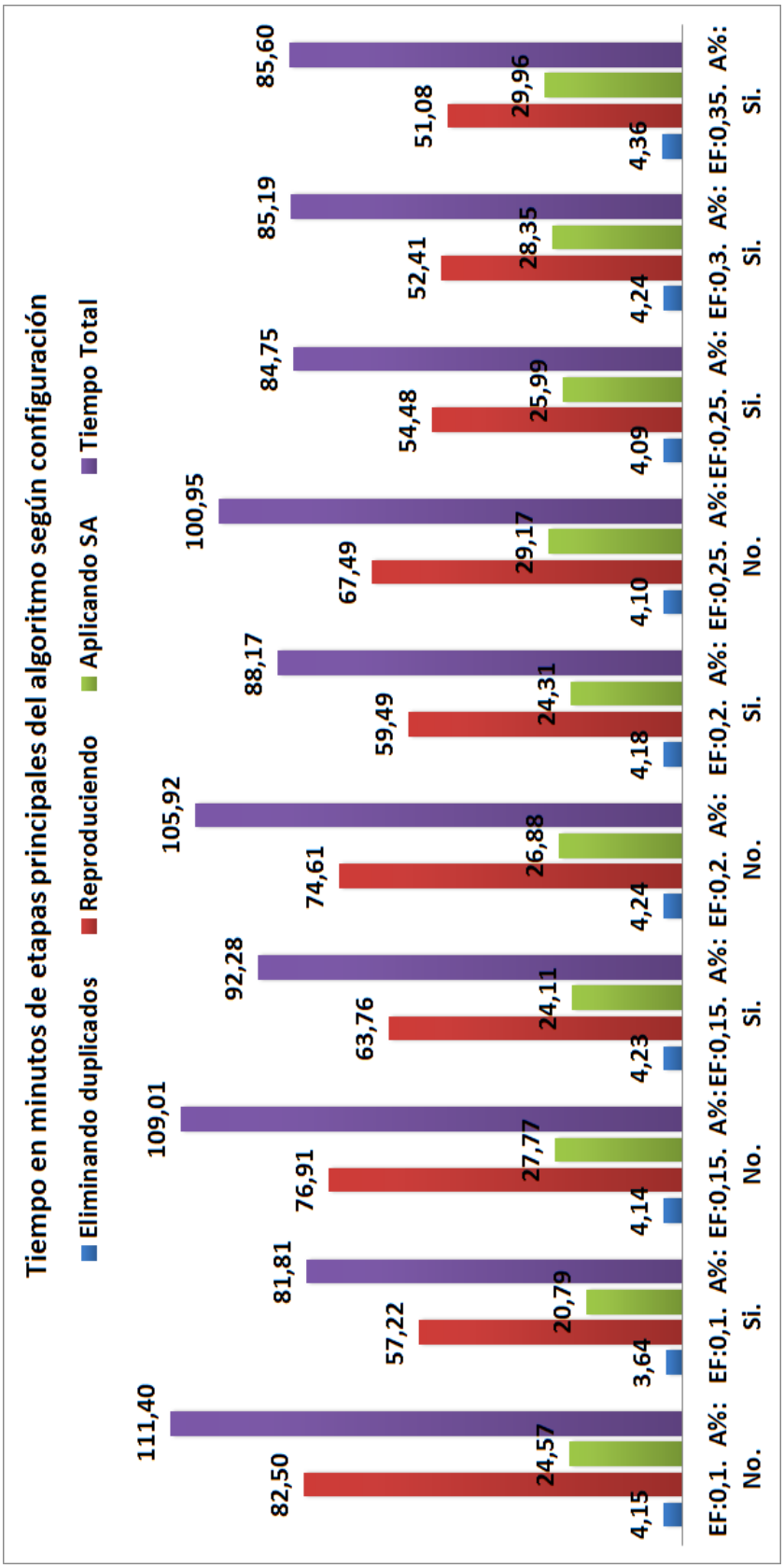

Figura 7.11: Tiempo en minutos de etapas principales del algoritmo según configuración. Colegio Saint Andrews

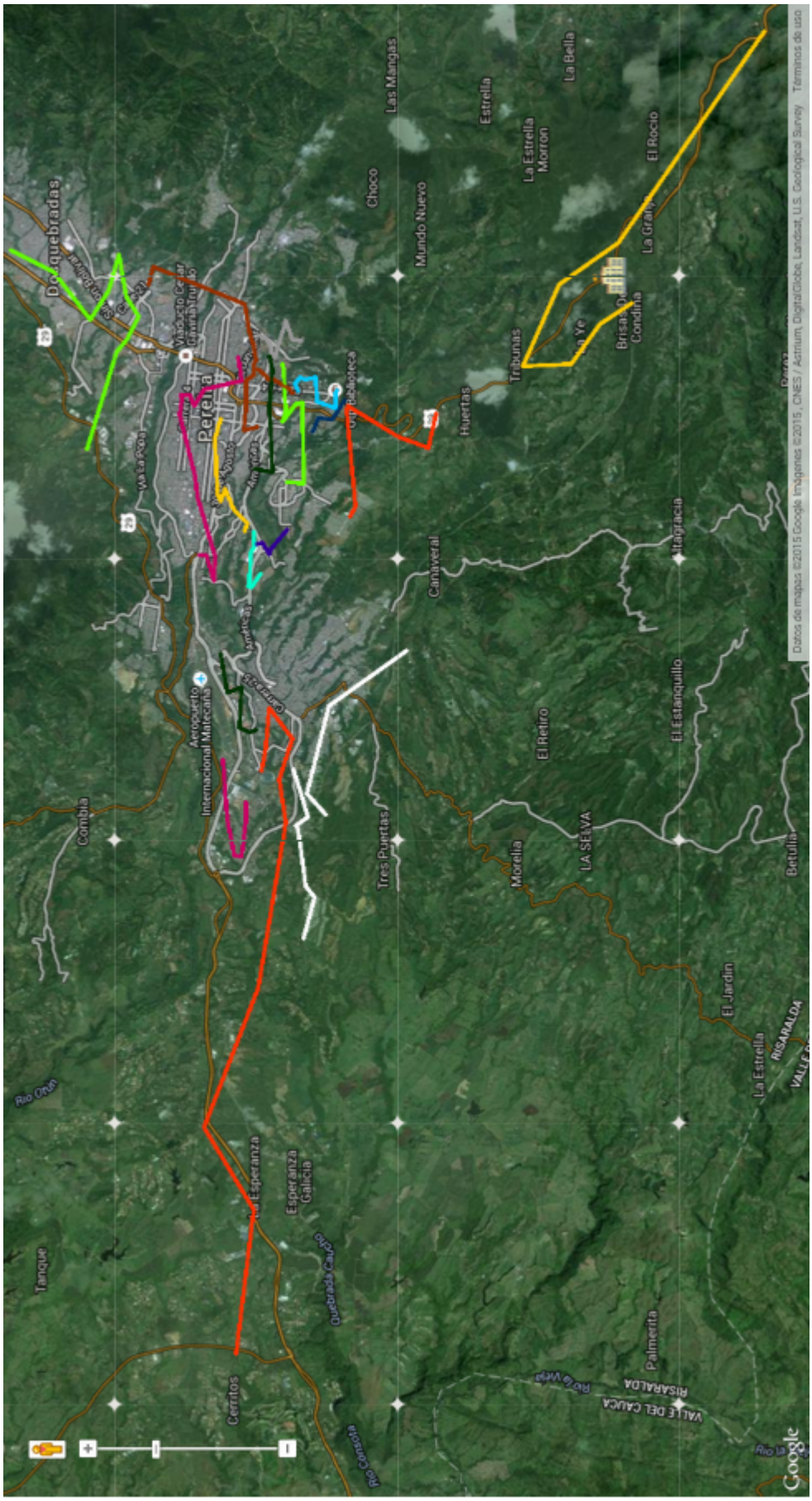

Figura 7.12: Mejor resultado encontrado Liceo Taller San Miguel, Bachillerato

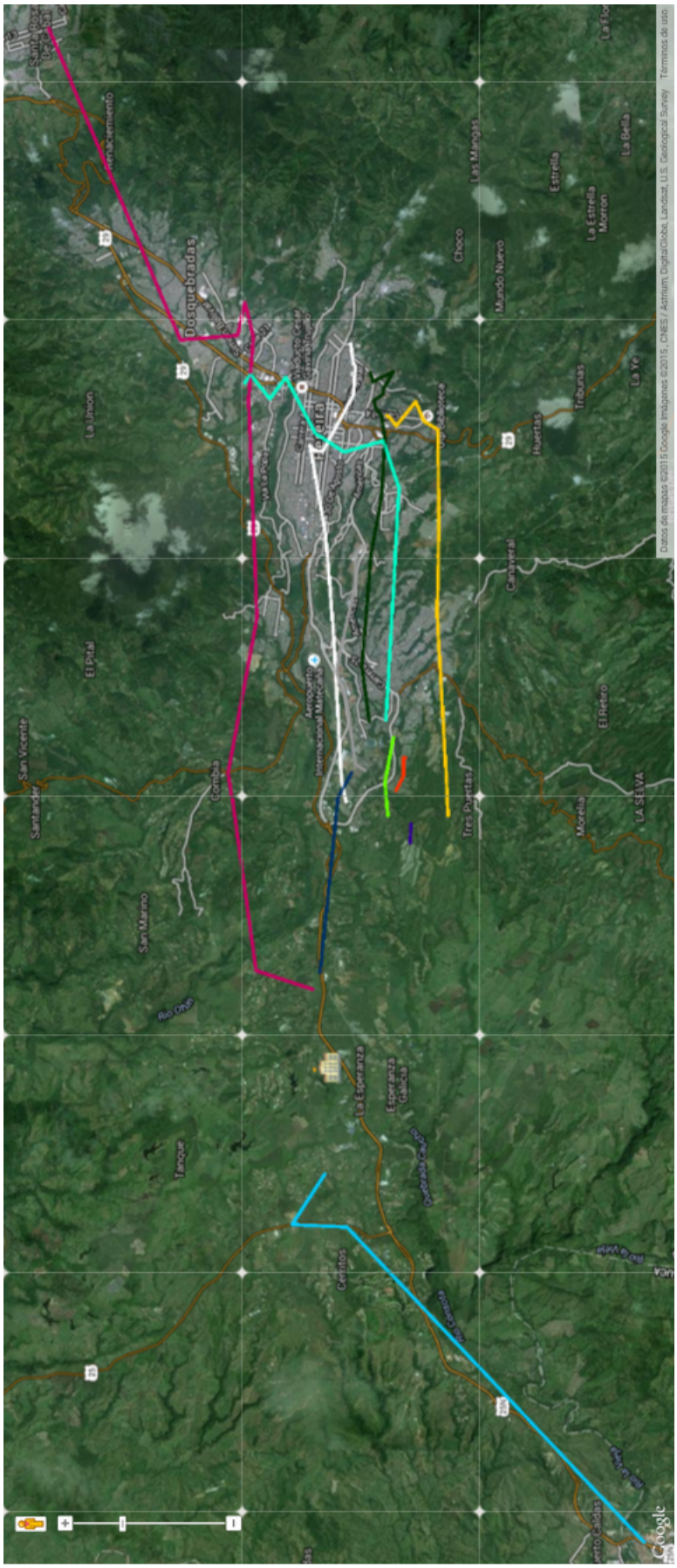

Figura 7.13: Mejor resultado encontrado. Colegio Saint Andrews

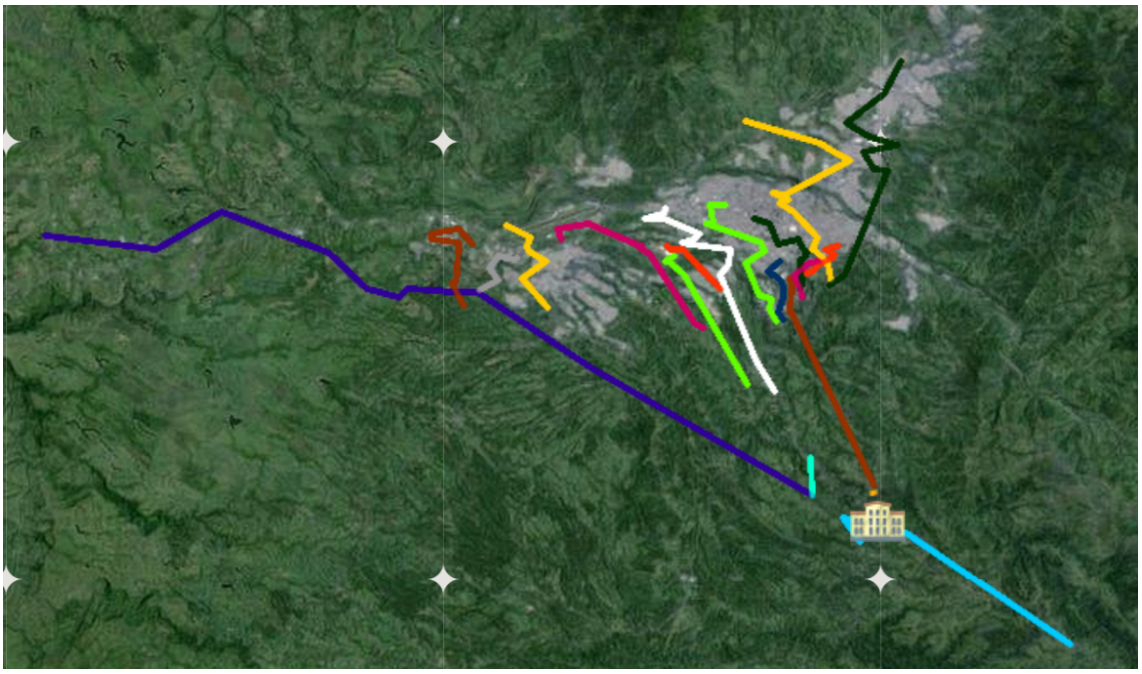

Figura 7.14: Gráfico Google Maps de Resultados Recocido Simulado, Liceo Taller San Miguel, Bachillerato. Sin las *Etiquetas* de los estudiantes.

En las figuras 7.12, 7.13 y 7.14 se observa como el software muestra en *google maps* el mejor resultado de Bachillerato del Liceo Taller San Miguel, con el algoritmo de recocido simulado.

7.3. Comparación de algoritmos

- Con las configuraciones y variantes probadas el algoritmo de recocido simulado es muy superior al genético tanto en tiempo como en resultado.
- Es de gran importancia poner una condición de parada al recocido simulado si se deja de encontrar una mejor solución tras una cantidad de iteraciones ya que la probabilidad de mejorar es muy baja y se tienen amplios beneficios en tiempo.
- El recocido simulado otorga mejores resultados en la medida en que se toma el *CoolingRate* mas bajo.
- La configuración ideal del algoritmo genético varía según el tipo de mapa. Debido a la gran variedad de posibles configuraciones del genético no se descarta que entre estas haya una combinación que produzca mejores resultados que los actuales, esto puede ser un problema mientras no se tengan definidos los criterios para decidir con que configuración resolver cada instancia, pero a su vez le da mayor flexibilidad al algoritmo.
- Aunque el recocido simulado de mejores resultados si se corre por si solo la solución sigue estando lejos de la mejor encontrada en el total de las pruebas, en los 3 mapas probados esta siempre se encontró corriendo primero el algoritmo genético y aplicando el recocido simulado a la mejor solución encontrada

por el primero, esta es la configuración recomendada y los buenos resultados indican que el recocido simulado se puede utilizar como un algoritmo de mejora, mientras que el genético hace mejor la tarea de agrupar. Un ejemplo de esto se dio en el mapa de LTSM, bachillerato, donde se logró una distancia total de 102km, mientras que ejecutándolo solo con el recocido simulado solo fue posible alcanzar 146.51km.

- Con la estrategia recomendada la configuración inicial del algoritmo genético no tiene influencia en el resultado final, lo único importante es entregarle al recocido simulado una solución avanzada y no cruda.

- Si bien las distancias que se utilizaron fueron geográficas y no teniendo en cuenta el recorrido por las vías, los algoritmos prueban encontrar soluciones que serían imposibles de hallar de forma manual. Debido a que el algoritmo no depende en lo absoluto de como se calculen las distancias sino su valor, si el programa se alimenta con las distancias reales las soluciones encontradas estarían listas para ser utilizadas por los colegios.

Apéndice A

Anexo

Nombre	Dirección	Zona	Grados	Presentaron Saber 11 (2012)	Presentaron Saber 11 (2013)	Estimación
Col Anglo-americano	Km 4 Vereda Huertas	Rural	-1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	30	48	507
Col Calasanz	Cr 19 No 46-50	Urbana	-1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	71	No se halló	923
Col Comercial Adoratrices	Kr 7 A 31 36	Urbana	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	29	37	396
Col Compañía De María La Enseñanza	Av Circunvalar 1-57	Urbana	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	25	30	385
Col Cooperativo Pereira	Cr 25 77-18 Corales	Urbana	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	65	76	846
Col Corporación Pino Verde	La Cadena El Tigre - Vereda Los Planes	Rural	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	14	21	245
Col De Los Sagrados Corazones	Km 8 Via Armenia	Rural	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	31	18	343
Col Del Inmaculado Corazón De María	Via Mundo Nuevo	Rural	-1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	35	38	474
Col Fundación Lic Ingles	Km 5 Via Cerritos	Rural	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	30	30	420

Col General Rafael Reyes	Km 11 Via A Cerritos	Rural	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	33	42	450
Col La Anunciación	Kr 3 19 38	Urbana	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	22	52	518
Col La Salle	Km 6 Via Cartago	Rural	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	134	139	1911
Col Lic Francés De Pereira	Via Armenia Km 5	Rural	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	40	36	532
Col Sagrado C. De J.Bethlemitas	Vda El Congolo Via Altagracia	Rural	-1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	28	No se halló	364
Col Saint Andrews	Km 10 Cerritos Teléfonos	Rural	-1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	13	6	123
Fundación Gimnasio Pereira	Kr 13 3 E 99	Urbana	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	31	41	504
Lic Bilingüe EL POBLADO	Mz 20 Cs 22 23 24 Poblado Ii	Urbana	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	12	16	196
Lic Campestre De Pereira	Km 11 Entrada 10 San Isidro	Rural	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	19	8	148
Lic Merani	Via Cerritos Km 10	Rural	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	20	13	231
Liceo Taller San Miguel	Km 8 Vía Armenia	Rural	-2, -1 , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	34	30	448

Cuadro A.1: Estimación de estudiantes de principales colegios privados

Resultado Promedio	**Tiempo (Minutos)**	**Configuración**		
		Temperatura Inicial	**Iteraciones en temperatura**	**Porcentaje de Enfriamiento**
170.81	371.150	1.0	5	1e-05
177.14	36.653	5.0	2	1e-05

177.14	36.653	5.0	2	1e-05
178.45	31.834	1.0	2	1e-05
179.11	30.280	1.0	1	1e-05
179.50	46.825	10.0	2	1e-05
179.71	45.690	10.0	5	1e-05
179.78	43.196	10.0	1	1e-05
180.53	38.991	5.0	1	1e-05
182.33	33.906	1.0	5	1e-05
182.76	36.038	5.0	5	1e-05
200.13	6.399	5.0	1	0.0001
201.59	7.472	10.0	1	0.0001
202.86	9.888	10.0	5	0.0001
209.50	5.413	1.0	2	0.0001
210.22	8.066	5.0	2	0.0001
211.18	6.808	5.0	5	0.0001
212.90	6.763	1.0	5	0.0001
218.06	6.865	1.0	1	0.0001
218.21	8.523	10.0	2	0.0001
256.03	0.791	5.0	1	0.001
257.09	0.758	1.0	2	0.001
257.11	0.870	1.0	5	0.001
257.75	1.036	10.0	5	0.001
260.24	0.913	10.0	2	0.001
265.62	0.963	5.0	5	0.001
266.42	0.848	5.0	2	0.001
268.95	0.816	10.0	1	0.001
270.55	0.711	1.0	1	0.001
340.34	0.082	5.0	2	0.01
368.68	0.080	10.0	2	0.01
373.37	0.092	5.0	5	0.01
377.92	0.068	1.0	2	0.01
386.38	0.080	5.0	1	0.01
387.12	0.083	10.0	1	0.01
390.70	0.067	1.0	5	0.01
395.66	0.068	1.0	1	0.01
397.29	0.071	10.0	5	0.01

Cuadro A.2: Resultados Recocido Simulado, Liceo Taller San Miguel, Primaria

Ejecutado con configuración: Temperatura Inicial: 1.0, Iteraciones por Temperatura: 5, Proporción de Enfriamiento: 1e-06, Número total de Repeticiones Ciclo Principal: 16118087		
Solucion	porcentaje	temperatura
314.550876312	1 %	0.852510221566
309.789691816	2 %	0.725604513099
286.324932804	3 %	0.617589525156
264.565386936	4 %	0.525653871629

245.815042679	5 %	0.447403949556
232.460644999	6 %	0.380802472657
221.329312521	7 %	0.324115429301
207.332882575	8 %	0.275866936416
203.513888161	9 %	0.23480081393
191.768015925	10 %	0.199848079187
188.131577207	11 %	0.170098280951
182.481735463	12 %	0.144777099185
173.261993593	13 %	0.123225280887
172.456396015	14 %	0.104881710817
168.559400145	15 %	0.0892688025108
165.187827833	16 %	0.0759800640136
162.993137324	17 %	0.0646695146025
160.810478806	18 %	0.055042732003
153.112520142	19 %	0.0468489571211
151.619147259	20 %	0.0398749245079
150.448781644	21 %	0.0339390608078
147.545928044	22 %	0.0288868220499
147.224122834	23 %	0.0245866699986
146.963626237	24 %	0.0209266474719
146.730498374	25 %	0.0178114634652
146.644362877	26 %	0.0151600273502
146.630743475	27 %	0.0129032743368
146.611216575	28 %	0.0109824662427
146.600283576	29 %	0.0093475936125
146.563788274	30 %	0.00795609150202
146.548983562	31 %	0.00677173127252
146.546141471	32 %	0.00576367735533
146.545006698	33 %	0.00490568442833
146.540916487	34 %	0.00417541827767
146.540916487	35 %	0.00355385688055
146.540916487	36 %	0.00302482239803
146.540916487	37 %	0.0025745410823
146.540916487	38 %	0.00219128957416
146.511142086	39 %	0.00186508967785
146.510007313	40 %	0.0015874485725
146.510007313	41 %	0.00135113769609
146.510007313	42 %	0.00115000569529
146.510007313	43 %	0.000978813469958
146.510007313	44 %	0.00083310527321
146.510007313	45 %	0.000709087499868
146.510007313	46 %	0.000603531268662
146.510007313	47 %	0.000513688356261
146.510007313	48 %	0.000437219645543
146.510007313	49 %	0.000372134225195
146.510007313	50 %	0.000316737871815
146.510007313	51 %	0.000269587617389
146.510007313	52 %	0.000229456247316

146.510007313	53 %	0.00019529891596
146.510007313	54 %	0.000166226315567
146.510007313	55 %	0.000141481522574
146.510007313	56 %	0.000120420290623
146.510007313	57 %	0.000102494277202
Repeticiones totales: 46255560		
Run Time: 04:43:59		

Cuadro A.3: Prueba con porcentaje de ejecución.

Porc \EF - A %	**0,1 - No**	**0,1 - Si**	**0,15 - No**	**0,15 - Si**	**0,2 - No**	**0,2 - Si**
5 %	371,79	387,20	367,36	364,38	370,92	366,41
10 %	357,75	372,84	349,43	345,47	347,03	343,06
15 %	349,84	365,19	341,83	338,16	337,34	334,07
20 %	344,24	359,65	336,67	333,23	333,77	328,73
25 %	339,68	353,58	333,93	328,70	329,78	324,60
30 %	335,86	349,33	330,48	324,19	326,86	322,71
35 %	331,85	346,06	328,59	321,20	324,75	319,96
40 %	328,51	342,48	326,36	318,45	321,93	317,36
45 %	326,41	340,06	322,58	315,83	319,40	315,07
50 %	322,90	336,48	320,72	313,93	317,61	312,83
55 %	320,52	333,93	318,46	311,35	315,91	311,25
60 %	318,91	330,69	316,29	308,99	314,81	309,41
65 %	316,81	328,15	314,36	307,17	312,37	307,36
70 %	314,81	325,21	312,51	305,25	310,88	305,07
75 %	312,73	322,95	310,87	303,73	309,89	303,67
80 %	310,96	319,87	308,99	301,77	308,44	302,51
85 %	309,53	316,89	307,38	300,16	306,86	299,71
90 %	308,49	314,48	306,55	298,50	305,47	298,25
95 %	306,69	312,88	304,95	296,37	304,01	296,98
100 %	**305,69**	**311,22**	**303,37**	**294,75**	**302,97**	**295,66**
Tras aplicar SA	**108,04**	**109,42**	**108,80**	**109,46**	**109,58**	**108,12**
Mejora	**35 %**	**35 %**	**36 %**	**37 %**	**36 %**	**37 %**

Cuadro A.4: Promedio de distancia total por porcentaje según configuración. Liceo Taller San Miguel, Bachillerato

Porc \EF - A %	0,25 - No	0,25 - Si	0,3 - No	0,3 - Si	0,35 - Si	0,4 - Si
5 %	368,74	368,37	374,00	373,30	386,93	396,91
10 %	343,66	343,51	338,83	342,69	344,69	353,90
15 %	335,09	333,23	328,38	332,41	328,77	336,82
20 %	329,84	328,55	321,62	324,55	319,06	326,53
25 %	326,37	324,43	318,07	320,46	314,82	320,19
30 %	323,39	320,55	316,86	319,26	310,73	317,77
35 %	320,66	317,45	313,53	315,46	306,60	315,00
40 %	317,90	315,34	311,19	314,06	304,54	312,02
45 %	316,16	313,50	308,47	311,51	302,35	309,05
50 %	314,56	312,04	307,73	310,40	301,40	307,76
55 %	313,79	311,02	306,66	309,38	299,30	305,94
60 %	312,16	309,28	305,13	307,34	297,51	305,00
65 %	310,56	307,16	303,19	306,49	295,60	303,56
70 %	309,46	305,14	302,43	304,86	294,67	302,18
75 %	307,42	303,86	301,85	302,97	293,66	300,79
80 %	306,28	301,92	301,01	301,60	291,94	299,36
85 %	305,06	300,90	299,40	300,20	291,10	298,32
90 %	303,87	299,21	298,85	299,04	289,43	297,16
95 %	302,45	298,25	297,59	297,79	287,70	295,71
100 %	**301,37**	**297,13**	**296,08**	**295,79**	**286,73**	**294,24**
Tras aplicar SA	**109,13**	**110,37**	**110,12**	**108,36**	**108,20**	**110,39**
Mejora	**36 %**	**37 %**	**37 %**	**37 %**	**38 %**	**38 %**

Cuadro A.5: Promedio de distancia total por porcentaje según configuración. Liceo Taller San Miguel, Bachillerato

Etapa \EF - A %	0,1 - No	0,1 - Si	0,15 - No	0,15 - Si	0,2 - No	0,2 - Si
Elim. duplicados	6,6	5,9	6,8	6,7	6,8	6,7
Reproduciendo	183,9	123,0	178,4	132,7	168,8	125,3
Aplicando SA	46,0	42,5	47,0	48,9	48,9	47,6
Tiempo Total	**236,7**	**171,7**	**232,5**	**188,6**	**224,9**	**179,8**

Etapa \EF - A %	0,25 - No	0,25 - Si	0,3 - No	0,3 - Si	0,35 - Si	0,4 - Si
Elim. duplicados	7,1	7,0	5,3	5,4	7,4	8,1
Reproduciendo	164,8	122,6	107,9	84,8	102,7	134,4
Aplicando SA	48,0	48,4	35,8	36,0	52,0	54,3
Tiempo Total	**220,2**	**178,3**	**149,3**	**126,5**	**162,4**	**154,5**

Cuadro A.6: Tiempo promedio en minutos de las etapas principales del algoritmo según configuración. Liceo Taller San Miguel, Bachillerato

Etapa \EF - A %	0,1 - No	0,1 - Si	0,15 - No	0,15 - Si	0,2 - No	0,2 - Si
Mejor al 100 %	281,96	270,14	272,04	277,39	280,69	278,86
Promedio al 100 %	305,69	311,22	303,37	294,75	302,97	295,66
Peor al 100 %	320,01	347,24	342,33	311,12	338,36	331,23
Mejor tras SA	102,04	103,03	102,38	103,57	105,48	103,97
Promedio tras SA	108,04	109,42	108,80	109,46	109,58	108,12
Peor tras SA	111,60	114,17	113,97	113,45	114,20	113,80

Etapa \EF - A %	0,25 - No	0,25 - Si	0,3 - No	0,3 - Si	0,35 - Si	0,4 - Si
Mejor al 100 %	284,58	271,32	279,47	274,11	262,98	271,40
Promedio al 100 %	301,37	297,13	296,08	295,79	286,73	294,24
Peor al 100 %	315,82	320,01	331,20	320,67	325,36	311,01
Mejor tras SA	105,14	104,38	106,45	106,00	102,33	107,24
Promedio tras SA	109,13	110,37	110,12	108,36	108,20	110,39
Peor tras SA	113,08	114,14	112,75	113,12	113,60	115,88

Cuadro A.7: Mejor, promedio y peor resultado tras aplicar el algoritmo genético y tras aplicar el recocido simulado, según configuración. Liceo Taller San Miguel, Bachillerato

Porc \EF - A %	**0,1 - No**	**0,1 - Si**	**0,15 - No**	**0,15 - Si**	**0,2 - No**	**0,2 - Si**	**0,25 - No**	**0,25 - Si**	**0,3 - Si**	**0,35 - Si**
5 %	229,61	224,92	219,57	226,84	229,17	225,57	225,51	227,89	236,65	243,46
10 %	214,94	212,19	205,69	208,50	212,81	202,98	207,57	207,57	212,44	218,58
15 %	208,62	205,60	200,34	202,51	205,68	195,08	200,12	199,36	203,05	209,85
20 %	204,50	202,30	196,01	196,56	200,12	192,42	194,92	196,07	195,62	204,50
25 %	200,15	199,21	194,35	193,16	196,12	189,22	192,95	191,89	192,40	200,74
30 %	198,10	197,18	192,26	190,08	194,07	186,92	190,25	189,93	190,27	199,02
35 %	195,65	194,67	190,57	186,75	191,78	184,69	188,47	188,14	186,93	196,21
40 %	194,63	192,43	189,66	185,68	189,74	181,44	186,59	186,12	184,57	191,41
45 %	193,13	191,75	188,11	184,12	187,41	180,19	184,75	184,83	182,38	189,74
50 %	192,13	190,70	187,24	183,20	185,32	178,13	183,54	183,30	179,63	188,81
55 %	191,73	189,50	185,96	182,03	184,49	177,17	182,26	181,48	178,84	187,02
60 %	191,22	189,05	185,79	180,60	183,59	175,98	180,60	179,50	177,26	185,28
65 %	189,89	188,15	185,15	180,28	181,69	175,07	180,18	178,39	175,84	183,45
70 %	189,19	187,28	184,28	178,97	180,64	173,92	179,35	177,67	174,15	181,90
75 %	188,61	186,06	183,62	178,20	180,23	172,93	178,36	176,70	172,60	180,23
80 %	187,64	185,70	183,45	178,12	179,21	171,98	177,87	176,08	171,55	179,49
85 %	186,86	185,19	183,07	177,66	179,07	171,02	176,97	175,15	170,86	177,90
90 %	186,47	184,57	182,57	177,07	178,75	170,14	176,60	174,40	169,98	176,46
95 %	185,76	184,22	181,84	176,75	178,31	169,26	176,47	173,94	168,73	175,22
100 %	**185,21**	**182,86**	**181,14**	**176,44**	**177,82**	**168,83**	**176,10**	**173,28**	**167,96**	**174,32**
Tras aplicar SA	**93,15**	**96,74**	**95,82**	**94,87**	**92,29**	**97,16**	**93,78**	**94,40**	**97,09**	**94,30**
Mejora	**50 %**	**53 %**	**53 %**	**54 %**	**52 %**	**58 %**	**53 %**	**54 %**	**58 %**	**54 %**

Cuadro A.8: Promedio de distancia total por porcentaje según configuración. Colegio Saint Andrews

Etapa \EF - A %	0,1 - No	0,1 - Si	0,15 - No	0,15 - Si	0,2 - No	0,2 - Si
Eliminando duplicados	4,15	3,64	4,14	4,23	4,24	139,72
Reproduciendo	82,50	57,22	76,91	63,76	74,61	168,83
Aplicando SA	24,57	20,79	27,77	24,11	26,88	188,86
Tiempo Total	111,40	81,81	109,01	92,28	105,92	90,56

Etapa \EF - A %	0,25 - No	0,25 - Si	0,3 - Si	0,35 - Si
Eliminando duplicados	4,10	4,09	4,24	4,36
Reproduciendo	67,49	54,48	52,41	51,08
Aplicando SA	29,17	25,99	28,35	29,96
Tiempo Total	100,95	84,75	85,19	85,60

Cuadro A.9: Tiempo promedio en minutos de las etapas principales del algoritmo según configuración. Colegio Saint Andrews

Etapa \EF - A %	0,1 - No	0,1 - Si	0,15 - No	0,15 - Si	0,2 - No	0,2 - Si
Mejor al 100 %	161,50	161,71	165,09	163,00	164,86	139,72
Promedio al 100 %	185,21	182,86	181,14	176,44	177,82	168,83
Peor al 100 %	209,07	203,01	190,47	190,65	198,94	188,86
Mejor tras SA	87,48	86,88	90,09	87,87	86,67	90,56
Promedio tras SA	93,15	96,74	95,82	94,87	92,29	97,16
Peor tras SA	97,76	104,68	101,59	106,00	100,70	101,51

Etapa \EF - A %	0,25 - No	0,25 - Si	0,3 - Si	0,35 - Si
Mejor al 100 %	163,61	153,27	158,78	161,24
Promedio al 100 %	176,10	173,28	167,96	174,32
Peor al 100 %	191,17	187,65	172,61	189,74
Mejor tras SA	87,91	87,42	90,10	85,21
Promedio tras SA	93,78	94,40	97,09	94,30
Peor tras SA	98,76	99,47	102,71	102,67

Cuadro A.10: Mejor, promedio y peor resultado tras aplicar el algoritmo genético y tras aplicar el recocido simulado, según configuración. Colegio Saint Andrews

Bibliografía

[1] FENTON Aish. "The Bees Algorithm for the Vehicle Routing Problem". En: *Computer Science* (2011).

[2] WREN Anthony y HOLLIDAY Alan. "Computer scheduling of vehicles from one or more depots to a number of delivery points". En: *Operations Research Quarterly Volumen 23* (1972), págs. 333-344.

[3] Programa de futuros científicos. *Colegios de Pereira [online]*. URL: `http://www.ofecfuturoscientificos.com/colegios-pereira.html`.

[4] ARBELAITZ Olatz; RODRIGUEZ Clemente y ZAMAKOLA Ion. "Low Cost Parallel Solutions for the VRPTW Optimization Problem. International Conference on Parallel Processing Workshops". En: *IEEE Computer Society* (2001), págs. 176-181.

[5] *Colegio La Salle [online]*. URL: `pereira.delasalle.edu.co/`.

[6] *Colegio Salesiano San Juan Bosco [online]*. URL: `http://www.salesianosjb.edu.co/`.

[7] RYAN David M.; HJORRING Curt y GLOVER Fred. "Extensions of the Petal Method for Vehicle Routing". En: *En Journal of the Operational Research Society* (1993), págs. 289-296.

[8] KIRKPATRICK S.; GELATT Jr C. D. y VECCHI M. P. "Optimization by Simulated Annealing". En: *Science, New Series, Vol. 220, No. 4598* (mayo de 1983), págs. 671-680.

[9] TARANTILIS Christos D. y KIRANOUDIS Chris T. "BoneRoute: An Adaptive Memory-Based Method for Effective Fleet Management". En: *Annalis of Operations Research, Volumen 115, Kluwer Academic Publishers* (2002), págs. 227-241.

[10] EILON Samuel; WATSON-GANDY Carl Donald y CHRISTOFIDES Nicos. *Distribution management: Mathematical modelling and practical analysis*. 1971.

[11] BELL John E. y MCMULLEN Patrick R. "Ant colony optimization techniques for the vehicle routing problem". En: *Department of Operational Sciences, Air Force Institute of Technology, Wright-Patterson AFB, OH, USA* (jul. de 2004).

[12] GAMBARDELLA Luca M.; RIZZOLI Andrea E. y DONATI Alberto V. "Ant Colony System for a Dynamic Vehicle Routing Problem". En: *Istituto Dalle Molle di Studi sull Intelligenza Articiale* (2005).

[13] GILLETT Billy .E. y MILLER Leland R. "A heuristic algorithm for the vehicle dispatch problem". En: *Operations Research Volumen 22 Tema 2* (1974).

[14] GAMBARDELLA Luca M.; TAILLARD Eric. y AGAZZI Giovanni. "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows". En: *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale* (1999).

[15] BULLNHEIMER Bernd; HARTL Richard F. y STRAUSS Christine. "Applying the Ant System to the Vehicle Routing Problem". En: *2nd International Conference on Metaheuristics,* (1997).

[16] GLOVER Fred. "Future Paths for Integer Programming and Links to Artificial Intelligence". En: *Computers and Operations Research, Volumen 13* (1986), págs. 533-549.

[17] CLARKE G y WRIGHT J. W. "Scheduling of vehicles from a central depot to a number of delivery points". En: *Operations Research Volumen 12 Tema 4* (1964).

[18] LAPORTE Gilbert. "The Vehicle Routing Problem: An overview of exact and approximate algorithms". En: *Centre de Recherche sur les Transports* (1992).

[19] LAPORTE Gilbert y NOBERT Yves. "Exact algorithms for the vehicle routing problem". En: *MARTELLO, Silvano et al. Surveys in Combinatorial Optimization* (1987).

[20] BEKTAS Tolga; ERDOGAN Gunes y ROPKE Stefan. "Formulations and Branch and Cut Algorithms for the Generalized Vehicle Routing Problem". En: *Transportation Science* (2011).

[21] HOLLAND John Henry. "Adaptation in natural and artificial systems". En: *University of Michigan Press* (1975).

[22] *ICFES Interactivo: Clasificación por planteles [online].* URL: `http://www.icfesinteractivo.gov.co/Clasificacion/`.

[23] CZECH Zbigniew J. y CZARNAS Piotr. "Parallel simulated annealing for the vehicle routing problem with time windows". En: *10th Euromicro Workshop on Parallel, Distributed and Network-based Processing* (ene. de 2001), págs. 376-383.

[24] DESROCHERS Martin; DESROSIERS Jacques y SOLOMON Marius. "A new optimization algorithm for the vehicle routing problem with time windows". En: *Operations Research Volumen 40 Tema 2* (1992).

[25] BERGER Jean y BARKAOUI Mohamed. "A hybrid genetic algorithm for the capacitated vehicle routing problem". En: *Genetic and Evolutionary Computation—GECCO 2003, volume 2723 of Lecture Notes in Computer Science* (2003), págs. 646-656.

[26] FISHER Marshall L. y JAIKUMAR Ramchandran. "Generalized Assignment Heuristic for Vehicle Routing". En: *Networks Volumen 2 Tema 2* (2008), págs. 109-124.

[27] *Liceo Taller San Miguel [online].* URL: `http://www.liceotallersanmiguel.edu.co/`.

[28] BALINSKI Michel Louis y QUANDT Richard Emeric. "On an integer program for a delivery problem". En: *Operations Research, Volumen 2 Tema 2* (1964).

[29] SUTHIKARNNARUN Nanthi. "A Sweep Algorithm for the Mix Fleet Vehicle Routing Problem". En: *Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol II IMECS* (2008).

[30] Networking y Emerging Optimization Research Group. *Solution Methods for VRP [online]*. URL: `http://neo.lcc.uma.es/vrp/solution-methods/`.

[31] CHRISTOFIDES Nicols y EILON Samuel. "An algorithm for the vehicle dispatching problem". En: *Operational Research Society, Volumen 20* (1969), págs. 309-318.

[32] KELLY James P. y XU Jiefeng. "A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem". En: *Transportation Science, Volumen 30* (1996), págs. 379-393.

[33] OTH Paolo y VIGO Daniel. "The Granular Tabu Search and its Application to the Vehicle Routing Problem". En: *Computers and Operations Research, Volumen 13* (1986), págs. 533-549.

[34] TOTH Paolo y VIGO Daniel. *The Vehicle Routing Problem*. 2002.

[35] NAGATA Yuichi. "Edge assembly crossover for the capacitated vehicle routing problem". En: *Evolutionary Computation in Combinatorial Optimization, Lectures Notes in Computer Science, Volumen 4446* (2007), págs. 142-153.