

MODELO DE OPTIMIZACION
PARA VEHICULOS DE
TRANSPORTE PÚBLICO COLECTIVO URBANO

JORGE ANDRES QUINTERO T.

Requisito para optar al título de
Magíster

Directora: **PhD. Patricia Jaramillo A.**

Universidad Nacional de Colombia
Sede Manizales
Maestría en Automatización Industrial
2005

RESUMEN

El presente trabajo es el resultado de varios años de análisis e investigación sobre la problemática del Transporte Público Colectivo Urbano en Colombia. Este servicio presenta graves deficiencias que afectan a los usuarios, conductores, otros tipos de vehículos, propietarios, entes reguladores y en general a toda una ciudad. La operación del sistema no es óptima y los procesos de planeación y control son inadecuados y en algunos casos inexistentes.

Uno de los factores primordiales para una buena operación del servicio, es la planeación del mismo, especialmente la programación de los rodamientos o itinerarios que deben cumplir los vehículos de Transporte Público Colectivo Urbano, para satisfacer adecuadamente la demanda de pasajeros, minimizar los gastos de operación y maximizar las ganancias de los conductores y transportadores.

Esa programación se hace normalmente en forma manual, basada en la experiencia del despachador, sin ajustarse a la demanda de pasajeros y completamente ajena a la optimización del servicio. Aunque existe un método clásico de programación manual, que busca obtener una programación con el mínimo número de vehículos, no se utiliza en la mayoría de las empresas, debido a la gran sobreoferta de vehículos. Este es un problema complejo debido a la cantidad de posibilidades y a que existen varios objetivos que se desean alcanzar, pero que son conflictivos entre sí. No fue posible aplicar los modelos de optimización para Transporte Público Colectivo Urbano desarrollados en otros países, dado que las condiciones de operación y las características propias del Transporte en Colombia son completamente atípicas a las del resto del mundo. Se propone entonces un modelo que da cuenta de esas características y se optimiza mediante un algoritmo genético híbrido como técnica heurística de optimización multiobjetivo.

AGRADECIMIENTOS

Especialmente, quiero dar las gracias a mi directora de tesis Dra. Patricia Jaramillo Álvarez, Ph. D. por su colaboración y paciencia. Sin su supervisión, su ayuda invaluable y sus conocimientos técnicos en el área de optimización, no habría sido posible realizar esta tesis de maestría.

También quiero hacer un homenaje póstumo al Arquitecto Mauricio Arcila Rivera, quien desde la Secretaría de Tránsito y Transportes de Manizales impulsó el desarrollo de alternativas tecnológicas en pro de lograr un Transporte Público Colectivo Urbano moderno, eficiente y rentable para la ciudad. Fue una fuente de ideas excelentes y la persona que me introdujo en la problemática del Transporte Público. Seguramente, estaría feliz de ver en la práctica la implementación de este modelo, pero lastimosamente, algunos violentos e intolerantes decidieron poner fin a su vida mediante las balas, aunque no a los proyectos que decididamente lideró.

Finalmente quiero agradecer a mi esposa, Erika Bibiana Pérez, por su compañía y apoyo, para la realización de este proyecto.

TABLA DE CONTENIDO

Resumen	I
Agradecimientos	II
Tabla de contenido	III
Lista de tablas	V
Lista de figuras	VII
Lista de anexos	VIII
Introducción	1
CAPITULO 1. Antecedentes	3
1.1 SOLUCIONES AL PROBLEMA DE ASIGNACIÓN DE CONDUCTORES.	7
1.1.1 <i>heurísticas tempranas.</i>	7
1.1.2 <i>métodos matemáticos.</i>	8
1.1.3 <i>métodos meta-heurísticos.</i>	8
1.2 OTROS PROBLEMAS RELACIONADOS	12
CAPITULO 2. Definición del problema	14
CAPITULO 3. Programación de rodamientos o itinerarios para vehículos de transporte público colectivo	27
3.1 IMPORTANCIA DE LOS RODAMIENTOS O ITINERARIOS.	27
3.2 ELEMENTOS BÁSICOS DE UN RODAMIENTO.	29
3.2.1 <i>horario de servicio.</i>	29

3.2.2	<i>intervalo.</i>	29
3.2.3	<i>frecuencia de servicio.</i>	29
3.2.4	<i>tiempo de recorrido.</i>	29
3.2.5	<i>tiempo de terminal</i>	29
3.3	ELABORACIÓN MANUAL DE RODAMIENTOS O ITINERARIOS (MÉTODO CLÁSICO).	30
3.4	ELABORACIÓN DEL RODAMIENTO O ITINERARIO PARA LA RUTA PILOTO.	33
CAPITULO 4. Introduccion a los algoritmos geneticos		38
4.1	EVALUACIÓN DE LA POBLACIÓN	40
4.2	MECANISMOS DE SELECCIÓN	41
4.3	CRUZAMIENTO	42
4.4	MUTACIÓN	42
4.5	EL ALGORITMO GENÉTICO SIMPLE (AGS)	42
4.6	FUNCIONES DE PENALIZACIÓN	44
4.7	OPTIMIZACIÓN DE MÚLTIPLES OBJETIVOS	46
4.8	VENTAJAS Y DESVENTAJAS DE LOS ALGORITMOS GENÉTICOS	48
CAPITULO 5. Algoritmo Genético para la programación de rodamientos o itinerarios		51
5.1	ESQUEMA DE CODIFICACIÓN	51
5.2	POBLACIÓN INICIAL	55
5.3	OPERADORES GENÉTICOS	57
5.4	EVALUACIÓN DE LA POBLACIÓN	58
5.5	ALGORITMO GENÉTICO PARA LA PROGRAMACIÓN DE RODAMIENTOS	64
5.6	ASIGNACIÓN DE LA PROGRAMACIÓN SEMANAL	64
CAPITULO 6. Casos de aplicación y resultados		68
6.1	PROGRAMACIÓN MANUAL DE LA RUTA PILOTO.	68
6.2	RESULTADOS DE LA PROGRAMACIÓN CON EL ALGORITMO GENÉTICO.	74
6.3	RESULTADOS DE LA PROGRAMACIÓN SEMANAL.	83
Conclusiones y perspectivas futuras		86
	TRABAJOS FUTUROS	88
Referencias		89

LISTA DE TABLAS

Tabla 1. Número de vehículos por empresa _____	18
Tabla 2. Ejemplo de tabla de despacho para la programación de 6 vehículos _____	31
Tabla 3. Ejemplo de tabla de despachos sacando vehículos a descansar _____	32
Tabla 4. Intervalos de despacho para la ruta piloto _____	34
Tabla 5. Tabla de despachos para un día _____	35
Tabla 6. Programación de una rutina _____	36
Tabla 7. Codificación del algoritmo genético _____	52
Tabla 8. Ejemplo de un cromosoma _____	53
Tabla 9. Rutinas asignadas y disponibles para cada despacho _____	54
Tabla 10. Resultados de la programación manual con tiempo de terminal. _____	69
Tabla 11. Resultados de la programación manual sin tiempo de terminal. _____	71
Tabla 12. Programación de una rutina sin tiempo de terminal _____	72
Tabla 13. Resultados del fitness con diferentes valores para los parámetros _____	75
Tabla 14. Resultados de la programación con el algoritmo genético. _____	75
Tabla 15. Programación de almuerzos y descansos. _____	76
Tabla 16. Resultados de la programación con diferentes métodos. _____	77
Tabla 17. Comparación de resultados utilizando diferentes criterios. _____	80
Tabla 18. Comparación de resultados utilizando diferentes criterios (normalizada). _____	81

Tabla 19. Programación semanal sin preferencias.	84
Tabla 20. Cálculo de las preferencias.	85
Tabla 21. Programación semanal con preferencias.	85

LISTA DE FIGURAS

Figura 1. Problemática actual del transporte.	14
Figura 2. Esquema de funcionamiento actual.	16
Figura 3. Pasajeros transportados por vehículo al día (2001).	17
Figura 5. Principal terminal de despacho de la empresa Socobuses S.A.	20
Figura 6. Caseta de despacho terminal principal de la empresa Socobuses S.A.	21
Figura 7. Función de pertenencia de un criterio	59
Figura 8. Comparación de las diferentes programaciones.	82

LISTA DE ANEXOS

Anexo 1. Programación clásica manual con tiempo de terminal.....	92
Anexo 2. Programación manual con tiempo de terminal ordenada por rutinas.....	94
Anexo 3. Programación clásica manual sin tiempo de terminal.....	96
Anexo 4. Programación clásica sin tiempo de terminal ordenada por rutinas.	98
Anexo 5. Programación clásica manual con 22 vehículos.....	100
Anexo 6. Programación manual con 22 vehículos ordenada por rutinas.....	102
Anexo 7. Programación con algoritmos genéticos.....	105
Anexo 8. Programación con algoritmos genéticos ordenada por rutinas.	107
Anexo 9. Código fuente del programa.....	109
Anexo 10. Código fuente de la programación semanal sin preferencias	117
Anexo 11. Código fuente de la programación semanal con preferencias	118

INTRODUCCIÓN

“El Transporte Urbano de pasajeros es un servicio primordial para toda ciudad; enlace entre las personas y el lugar en el que quieren estar. No produce bienes de consumo tangibles, pero hace posible que éstos se produzcan al trasladar diariamente a millones de trabajadores; no educa, pero lleva hasta los centros de estudio a miles de estudiantes; no proporciona diversión ni esparcimiento, pero apoya y hace posible el desarrollo de estas actividades” (Molinero y Sánchez, 1998).

En el Transporte Público intervienen procesos de planeación, diseño, operación, administración y control, pero unos de los más descuidados, por lo menos en Colombia, son los procesos de planeación, de los cuales, la programación de rodamientos o itinerarios forma parte importante. La realización eficiente de este proceso, aunque no es la solución a todos los problemas del transporte, puede contribuir en gran manera a que se tenga un servicio eficiente, moderno y organizado.

La programación actual de rodamientos es un proceso manual y ajeno a la optimización. Por lo tanto, la problemática que se aborda en esta tesis, es el desarrollo de un modelo de optimización para la programación de rodamientos en vehículos de Transporte Público Colectivo. Para dicha optimización se hace necesario explorar diversas técnicas, que sean capaces de realizar procesos de optimización sobre problemas con varios objetivos conflictivos entre sí.

Hay muy poca información sobre el proceso de programación de itinerarios, ya que todo se hace en forma empírica y aunque se han desarrollado modelos de optimización para Transporte Público Colectivo Urbano en otros países, se hace difícil adaptarlos a las condiciones de operación del Transporte Público en Colombia. Uno de los pocos trabajos previos realizados en el área de programación del servicio, se encuentra en (Molinero y Sánchez, 1988), aplicado al caso Mexicano, que es muy diferente al sistema Colombiano, y dice que la programación del servicio “... se trata

por primera vez en nuestro medio por lo que su aportación seguramente enriquecerá y promoverá un cambio en la forma que se opera el Transporte Público Urbano”.

El trabajo comprende el estudio de las características, componentes, restricciones y modo de operación actual del sistema, para lograr su caracterización. La investigación tecnológica que busca establecer una idea clara del estado del arte y de las opciones tecnológicas que se encuentran en el mercado y que pueden dar solución satisfactoria al problema en estudio. Seguidamente, se formula el modelo que describe la esencia del sistema y caracteriza sus variables. Teniendo el modelo inicial que describe el problema es necesario entonces escoger la técnica de solución que nos permite encontrar los valores de las diferentes variables controlables del sistema para dar solución al problema planteado. Existen dos alternativas básicas para enfrentar la solución de problemas de optimización: herramientas basadas en los principios de la programación matemática y herramientas heurísticas y metaheurísticas fundamentadas en la lógica, la imitación de sistemas y el buen razonamiento. Luego se implementa la solución computacional (algoritmo) y se aplica a casos concretos para depurar los errores y obtener un modelo final que arroje resultados confiables.

La tesis está organizada de la siguiente forma: en el capítulo 1 se presentan los antecedentes del problema y su diferencia con el problema de otros países, además de que se citan las técnicas de optimización que se han aplicado para su solución. El capítulo 2 hace un análisis crítico de los problemas del Transporte Público y su relación con la programación de rodamientos. En el capítulo 3 se explica detalladamente el proceso de programación de rodamientos e itinerarios y en el capítulo 4 se presentan algunas generalidades de los algoritmos genéticos. El capítulo 5 presenta el modelo y su optimización mediante un algoritmo genético híbrido. Los resultados que se obtienen con el modelo propuesto y con los métodos clásicos de programación se pueden observar en el capítulo 6. Finalmente, aparecen las conclusiones que se obtuvieron y los posibles trabajos futuros.

CAPITULO 1

ANTECEDENTES

Al hablar de programación de rodamientos o itinerarios para Transporte Público Colectivo Urbano es necesario mencionar que no existe mucha información teórica respecto al tema. Lo que existe está aplicado al modelo de transporte que se maneja en otros países, pero dadas las condiciones particulares que se presentan en Colombia, es difícil adaptar esa información a nuestro sistema. Tradicionalmente se ha manejado el Transporte Público desde un punto de vista empírico, regulado por normas emitidas por el Ministerio del Transporte, orientadas principalmente hacia los aspectos legales y las normas de tránsito.

Todo lo relacionado con planeación, programación del servicio y control sobre la operación está basado en la experiencia y no hay escritos procedimientos ni métodos. Además debido a que todos los procesos se manejan manualmente y se carece totalmente de tecnología para el despacho y control del Sistema, es difícil establecer antecedentes, por lo menos en Colombia.

El único texto en español que aborda el problema de la programación del servicio y la elaboración de itinerarios es el de Molinero y Sánchez (1988), el cual se atreve a afirmar que es la primera vez que se trata el tema en nuestro medio. Este autor propone dos métodos de programación de itinerarios: un método gráfico y un método analítico. Ambos métodos son manuales y tan complejos que haciendo un ejemplo de programación para una ruta utiliza desde la página 418 hasta la página 509 de su libro. Además el éxito de la programación en estos métodos depende de la experiencia del

programador en la elaboración de los itinerarios. Este modelo mexicano, al igual que los desarrollados en otros países no es fácilmente adaptable al caso colombiano.

Sin embargo, en este libro se definen elementos básicos importantes para realizar la programación de itinerarios, tales como: Intervalo, frecuencia de servicio, tiempo de recorrido y tiempo de Terminal.

El esquema bajo el cual opera el Transporte Público en otros países es muy diferente al nuestro. El transporte es un servicio público y por tal motivo en muchos países es subsidiado por el gobierno. Las empresas son realmente transportadoras, ya que son propietarias de todos los vehículos y contratan a los conductores para que cumplan su jornada laboral conduciendo diferentes vehículos de la empresa. Los conductores tienen un salario fijo que no depende de la movilización de pasajeros y la empresa se preocupa más por la calidad del servicio que por los ingresos económicos. Por el contrario en Colombia, la mayoría de las empresas de transporte, no son las propietarias de los vehículos; cada propietario tiene su conductor y le paga por porcentaje de acuerdo al movimiento de pasajeros. Este esquema de competencia desenfrenada genera un fenómeno conocido como la “*guerra del centavo*”, caracterizado porque los conductores luchan entre sí para movilizar la mayor cantidad de usuarios, sin importar los tiempos de recorrido, operando a velocidades peligrosas y pasando por alto la programación.

En otros países y particularmente en Europa, los Sistemas de Transporte Público cuentan con tecnología de punta (sistemas de posicionamiento global, computador en el vehículo, comunicación entre el centro de control y cada uno de los buses, cámaras de video en los vehículos, paraderos donde los usuarios pueden sentarse cómodamente y reciben información actualizada de los vehículos que vienen en camino, sistemas de pago electrónico, etc.), que garantizan una mejor prestación del servicio, pero que representan costos muy elevados que pueden no estar al alcance de todas las empresas de transporte colombianas.

En Estados Unidos el problema de la programación de rodamientos se conoce como “*Transit Scheduling*” (Programación de Tránsito) y se divide en cuatro etapas (Pine et al., 1998):

1. “*Trip generation*” (generación de viajes): Consiste en generar los viajes o recorridos para todo un día, teniendo en cuenta la frecuencia de despachos.
2. “*Blocking*” (generación de bloques): Es el proceso de asignación de vehículos y establece los recorridos que debe efectuar cada vehículo en un día.
3. “*Runcutting*” (Partición de rutinas): Se trata de hacer la asignación de conductores. Se toman los bloques y se dividen entre los conductores respetando la jornada laboral y dejándoles tiempo para almorzar y descansar.
4. “*Rostering*” (Generación de listados): En esta etapa se agrupan las rutinas diarias en paquetes de semanas o meses, teniendo en cuenta los días de descanso.

La Administración Federal de Tránsito de los Estados Unidos publicó unos manuales (Pine et. al., 1998) que explican a profundidad como se deben realizar estas cuatro etapas para lograr una programación eficiente del servicio. Sin embargo, tanto en Estados Unidos como en Europa se maneja el mismo modelo de operación: como todos los vehículos son propiedad de la empresa de transporte, el proceso de asignación de vehículos es muy simple, porque se trata de utilizar el menor número de vehículos posible. La parte complicada es la asignación de conductores, porque deben respetarse unas jornadas de trabajo y tiempos de descanso, entonces se asigna a un conductor para que conduzca un vehículo durante un tiempo, luego pasa a conducir otro y al terminar su jornada laboral se va para su casa. Pero el vehículo está programado para trabajar todo el día, solo que va a tener diferentes conductores.

En cambio en Colombia ese modelo es inaplicable, debido a que los vehículos no son propiedad de la empresa y el esquema de operación establece que cada vehículo tiene su conductor y este lo maneja durante todo el día. Ningún propietario permitiría que su vehículo fuera manejado por diferentes conductores. Entonces cuando se programa un vehículo se está programando con él al conductor y ya no se puede poner a trabajar el vehículo al máximo, porque hay que tener en cuenta los tiempos de descanso y almuerzo como si fueran parte del vehículo.

Se podría decir que la diferencia fundamental entre el modelo de transporte colombiano y los modelos de otros países, es que en Colombia se programan vehículos y en el resto del mundo se programan conductores. Es por esa razón que la

programación de itinerarios se maneja en el mundo como “*Bus driver scheduling problem*” (Problema de programación de conductores de buses). Es un problema mundial que lleva más de 40 años de investigación y es de interés porque la programación eficiente puede representar grandes ahorros económicos a las empresas de transporte público.

Como ya se mencionó se trata de partir bloques de trabajo, efectuados por un vehículo, en rutinas legales para cada uno de los conductores. El principal objetivo consiste en minimizar el número total de rutinas (“shifts”) y su costo.

Se establecen varios tipos de rutinas, por ejemplo hay rutinas formadas por dos piezas de trabajo separadas por un almuerzo que dura de 30 minutos a una hora. Y jornadas de aproximadamente 12 horas formadas por dos piezas de trabajo separadas por un descanso largo de varias horas. En Estados Unidos se acostumbra programar rutinas que tienen un solo bloque de trabajo sin almuerzo.

En Colombia podríamos decir que sólo existen tres etapas: generación de viajes, asignación de vehículos y programación semanal. Eso quiere decir que se combina la asignación de vehículos con asignación de conductores, aumentando las restricciones, complicando el problema y por ende complicando también el modelo.

Todo el proceso de programación de rodamientos o itinerarios se hace en forma manual y basado en la experiencia del encargado. Hay empresas que utilizan un método gráfico denominado “*Tabla Gantt*”, pero también es un método empírico que consiste en ir introduciendo vehículos para cubrir las horas de máxima demanda y luego irlos sacando de servicio a medida que la demanda disminuye. Es un método complejo porque cualquier variación implica construir de nuevo el gráfico. (Borndörfer et al., 2003).

Casi todos los problemas de transporte son problemas de decisión complejos por la variedad de variables que involucran, por la cantidad de beneficiarios que afectan (Usuarios, empresarios, conductores), y por el excesivo número de posibles soluciones (pero solo unas cuantas óptimas o eficientes), así que es casi imposible que se encuentre la mejor solución con un método manual. Por eso desde hace más

de 60 años se reconoció la necesidad de incluir procesos de optimización en los sistemas de transporte público (Borndorfer et. al., 1998).

Prueba de la importancia de la optimización en planeación del transporte es el hecho de que existan gremios como: EURO Working Group Transportation (EWGT) (<http://euro2005.cs.put.poznan.pl/>), Erasmus Center for Optimization in Public Transport (<http://www.few.eur.nl/few/research/ecopt/>) y otros.

Los antecedentes referenciados a continuación son todos sobre el “problema de asignación de conductores”. Tema en el que ha habido mucho interés desde el año 1960, aplicado a la programación de buses y también de trenes. Wren (1996) presenta una revisión histórica de las soluciones que se han planteado a este problema.

1.1 Soluciones al Problema de Asignación de Conductores.

El problema ha tratado de resolverse por diferentes métodos, pero debido a que no es el mismo problema del transporte en Colombia no se dan detalles respecto a cada una de esas soluciones. Sin embargo, los métodos utilizados se podrían agrupar cronológicamente de la siguiente forma:

1.1.1 Heurísticas Tempranas.

Fueron la primera aproximación al problema. Se basan en heurísticas porque los computadores no estaban muy desarrollados y los métodos matemáticos tampoco. El objetivo es buscar la mejor solución inicial posible y hacerle algunos refinamientos. Uno de los primeros sistemas que apareció fue TRACS (*Techniques for Running Automatic Crew Scheduling*) desarrollado en la Universidad de Leeds desde 1967 (Parker y Smith, 1981). También se desarrolló RUCUS (*RUn CUTting and Scheduling*) a finales de los años sesenta (Bodin et al., 1985). En los años setenta se desarrolló en Hamburgo HOT (*Hamburg Optimisation Techniques*) que fue muy popular en Alemania (Taha, 1997). COMPACS (*COMPUter Assisted Crew Scheduling*) fue desarrollado en los ochenta e incorporado al paquete de programación BUSMAN.

1.1.2 Métodos Matemáticos.

El problema de asignación de conductores puede formularse como un problema de programación lineal entera con dos modelos: “*Set Covering*” y “*Set partitioning*”, cuya formulación matemática se puede consultar en Li, (2002). El problema con estos modelos es que el número de rutinas es demasiado grande para encontrar una solución, por lo cual se combinan con heurísticas para reducir la magnitud del problema o se descompone en subproblemas más sencillos. A finales de los años 70 se desarrolló IMPACS (*Integer Mathematical Programming for Automatic Crew Scheduling*) instalado en Londres en 1984 (Wren y Smith, 1988). Luego apareció TRACS II como una nueva generación de software para programación de conductores. HASTUS (Blais y Rousseau, 1988) es un paquete comercial dividido en módulos que realiza todo el proceso de programación. En 1992 se presentó EXPRESS (Falkner y Ryan, 1992) un sistema desarrollado especialmente para Nueva Zelanda.

1.1.3 Métodos Meta-Heurísticos.

Nuevos investigadores han hecho una mezcla entre las técnicas de inteligencia artificial y métodos de investigación de operaciones para tratar de encontrar soluciones óptimas. Esto se hace necesario en vista de que la programación de itinerarios o rodamientos es un problema en el cual las variables de decisión deben definirse dentro de un espacio discreto, pero en el que el número de alternativas posibles es equivalente a todas las posibles combinaciones de todos los posibles valores de las variables. Estos problemas se han definido como problemas combinatoriales y en la última década se han realizado numerosas propuestas de métodos de optimización explícitos para este tipo de problema.

Los métodos para resolver problemas combinatoriales pueden clasificarse en exactos y heurísticos. Los exactos garantizan encontrar el óptimo global, pero hay muy pocas aplicaciones en la literatura, debido a su ineficiencia por la alta dimensionalidad que debe manejarse. Los heurísticos o meta heurísticos solo permiten aproximarse a los óptimos globales pero son más eficientes y flexibles.

Los métodos heurísticos siguen en general una de las dos estrategias básicas: “divide y vencerás” y mejora iterativa. Los métodos que siguen la primera, dividen

apropiadamente el problema en subproblemas de tamaño más manejable y luego juntan las subsoluciones. Los que siguen la segunda, comienzan con una configuración conocida del sistema. Se aplica luego una operación estándar de re-assignar a todas las partes del sistema en turno, hasta que se descubre una configuración que mejore la función objetivo. La configuración re-asignada se convierte en una nueva configuración del sistema y el proceso continúa hasta que no pueda obtenerse mejoras. Esta búsqueda puede quedarse en un óptimo local por eso es recomendable hacer el proceso varias veces comenzando desde diferentes configuraciones generadas aleatoriamente y salvar los mejores resultados.

Lo anterior, bastante complejo por la alta dimensionalidad de los problemas, se vuelve aún más, cuando se desean optimizar varios objetivos simultáneamente. A los problemas inherentes a la optimización multiobjetivo debe sumárseles un mayor costo y dificultad computacional.

Diferentes métodos exactos y meta heurísticos pueden adaptarse a varios problemas OCMO (*Optimización combinatorial multiobjetivo*) para generar soluciones eficientes, pero no siempre es fácil. El desarrollo de métodos para resolver problemas OCMO, es relativamente reciente. Algunos aportes básicos fueron:

Schaffer (1984) y Srinivas y Kalyanmoy (1994) propusieron métodos multiobjetivo basados en algoritmos genéticos.

Serafini (1992) ha propuesto algoritmos similares basados en recocido simulado y sugiere su uso en OCMO. La diferencia con los algoritmos clásicos de recocido simulado para un solo objetivo, fue el uso de reglas de agregación ponderada de valores de los objetivos.

Jaszkiewicz (1996) propuso también un método basado en recocido simulado que denominó "recocido simulado de Pareto". El método trata de combinar las ventajas de los métodos interactivos y los métodos con articulación de preferencias a posteriori. El método genera una muestra de soluciones aproximadamente eficientes con un procedimiento de recocido simulado multiobjetivo y luego hace un análisis del

conjunto de soluciones generado con un procedimiento interactivo para problemas con muchas alternativas finitas.

Algunos de esos métodos son:

- Recocido simulado (RS): La característica esencial de este método es perturbar una alternativa seleccionada aleatoriamente. Esta perturbación producirá una nueva alternativa que reemplazará a la anterior. El reemplazo será aceptado o no en función de alguna regla de decisión. El proceso se repite iterativamente hasta que se considere que se llegó al óptimo. El método se basa en el método "recocido simulado" realizado en mecánica estadística, para encontrar el límite inferior de energía de un material.
- Búsqueda Tabú: Método propuesto por Glover (1989) como una técnica iterativa que escapa de óptimos locales aceptando soluciones no tan buenas, y moviéndose paso a paso hacia una solución lo más cercana posible a un óptimo global. (Glover y Laguna, 1997). Se desarrolló HACS (Shen y Kwan, 2001) un sistema que aplica búsqueda Tabú al problema de la asignación de conductores. La búsqueda tabú, al contrario de otros métodos que tratan de escapar de los óptimos locales aleatoriamente, utiliza una búsqueda inteligente basada en formas sistemáticas.
- Algoritmos Genéticos: Se basan en los mecanismos de la genética y la selección natural. Se han propuesto algoritmos genéticos para resolver problemas de "set covering" y "set partitioning" (Chu y Beasley, 1995). Los han aplicado a la asignación de conductores (Kwan et al., 2001) y (Dias et al., 2001). Son aproximaciones al problema de asignación de conductores, pero para aplicarlo al Transporte Público Colombiano, es necesario hacer un modelo de asignación de vehículos. Se profundizará en este tema en el Capítulo 4.
- Colonias de hormigas: Los estudios sobre la hormiga muestran que colonias de hormigas logran determinar caminos cortos hacia fuentes de alimentos a través de la búsqueda semi-aleatoria de un grupo de hormigas que depositan feromonas. La feromona depositada constituye la información que permite la coordinación de los esfuerzos del enjambre en busca de comida: una hormiga

está más inclinada a seguir una trayectoria cuanto más fuerte sea el rastro de feromona depositada en ella. ¿Cómo ayuda esto a encontrar el camino más corto a la comida? Pues resulta que la hormiga que da con la comida primero (presumiblemente porque encontró un camino más o menos corto hacia el alimento) también es la primera en depositar el doble de feromona sobre una trayectoria al seguir ésta en su camino de vuelta a la colonia. Otras hormigas preferirán ligeramente esta trayectoria en comparación con otras que solo cuentan con la mitad de feromona en ellas, y al hacerlo intensifican el rastro de feromona haciendo la ruta aún más atractiva para nuevas hormigas. Como al principio el rastro de feromona no es muy fuerte, las primeras hormigas que siguen el camino hacia la comida pueden desviarse y encontrar atajos dentro del mismo, con lo que se perfecciona la solución propuesta. Eventualmente la mayor parte de hormigas siguen el camino hasta agotar la fuente de alimento. Wren (1997) aplicó este método a la programación de conductores.

- Programación con restricciones: Es una técnica para la descripción declarativa y solución efectiva de problemas combinatorios. Curtis (2000) la utiliza para la programación de Transporte Público.

Así que se puede hablar de dos aproximaciones al problema: la constructiva que abarca las heurísticas tempranas que buscan una solución inicial buena. Y por otro lado la de generar y seleccionar que comprende los métodos matemáticos y meta-heurísticos.

Con esta perspectiva histórica de las aproximaciones al problema de asignación de conductores, se empezó a buscar bibliografía más específica que permitiera comprender el modelo de otros países con las soluciones propuestas y de esa forma llegar a formular un modelo propio para el Transporte Público en Colombia.

La mayoría de los trabajos relacionados con el problema de asignación de conductores, lo resuelven con la estrategia de generar y optimizar. Primero se genera una gran cantidad de rutinas que no violen las restricciones y luego se seleccionan las mejores en un proceso iterativo. Debido a la cantidad de rutinas posibles se ha

utilizado el método de “Generación de Columnas” (“Column generation”) para resolver el problema (Fores, 1996).

Los modelos matemáticos más utilizados para resolver el problema de asignación de conductores son “*set covering*” y “*set partitioning*” y algunos autores han propuesto algoritmos genéticos como una buena forma de solucionar esos modelos (Beasley y Chu, 1994), en ellos se presentan las características del algoritmo, pero no se puede aplicar ese modelo fácilmente al problema colombiano. Esos dos modelos matemáticos suponen que se tienen todas las rutinas posibles y simplemente se trata de escoger las mejores para cubrir todos los despachos. En el caso colombiano lo más difícil es generar esas rutinas posibles.

También se han propuesto métodos heurísticos (Lourenco, 2000) de los cuales se destacan los algoritmos genéticos y la búsqueda tabú, como los que entregan mejores resultados.

1.2 Otros Problemas Relacionados

Muchos de los esfuerzos en el área de transporte se han orientado a la solución del problema de “ruteo de vehículos” (“*Vehicle Routing Problem*”), que consiste en buscar los recorridos óptimos en las rutas de transporte para optimizar los recursos. Este problema se presenta en las empresas de transporte de mercancías, que deben entregar y recoger productos en diferentes sitios y se trata de buscar cuál es la ruta óptima para esos recorridos. En Mauttone (2004) se presenta el estado del arte respecto a modelos y algoritmos para el problema de optimización de rutas. Hay trabajos como el de Machado et al (2000) que plantean el problema y lo resuelven utilizando algoritmos genéticos combinados con heurísticas. También Gruttner (2001) aborda el problema para generar nuevas rutas en sistemas de transporte público utilizando algoritmos genéticos. Kocjan (2001) reseña los métodos heurísticos que se pueden aplicar a este problema, aplicándolo a trenes. Aunque este no es el problema que se está tratando de resolver, las soluciones que se plantean dan una luz de posibles métodos que se adaptarían al problema de programación de vehículos.

El otro problema posterior a la programación de los despachos para un día es la asignación de la programación para una semana o un mes. Este proceso se conoce como “*Crew scheduling*” o “*rostering*” y también se han propuesto diferentes soluciones a este problema (Caprara, 1995), tales como Grobner (1998) que utiliza un algoritmo genético híbrido.

Normalmente, se hace primero la programación y después la asignación semanal, pero algunos autores están trabajando en la combinación de esos procesos (Wren, 1997), (Huisman, 2004).

El problema de programación de vehículos en Transporte Público puede tener similitudes con otros problemas de asignación, tales como asignación de exámenes, asignación de aulas, etc. Este problema se resuelve con algoritmos genéticos en la tesis de doctorado de Fang (1994).

Después de esta revisión de antecedentes se observa que plantear un modelo matemático para el problema de la programación de itinerarios de vehículos de transporte público en Colombia, es muy complejo, debido a las restricciones particulares del sistema. Se observa gran interés de los investigadores por aplicar métodos heurísticos, en particular algoritmos genéticos para problemas con restricciones tan fuertes y con múltiples objetivos. Los algoritmos genéticos no encuentran la mejor solución de todas, pero son capaces de entregar soluciones apropiadas en un tiempo inferior o comparable a los métodos matemáticos convencionales.

Hay una nueva tendencia en los algoritmos genéticos que se llama MOGA (*Multi-Objective Genetics Algorithms*) que trata de utilizar algoritmos genéticos para resolver problemas de múltiples objetivos (Coello, 2002). Se estudiarán este tipo de algoritmos para diseñar uno apropiado al problema en cuestión.

En los siguientes capítulos se explicarán las características del transporte público en Colombia y las restricciones que lo hacen diferente al problema de asignación de conductores que se maneja en otros países.

CAPITULO 2

DEFINICIÓN DEL PROBLEMA

El Transporte Público Colectivo Urbano en un país en vía de desarrollado es muy diferente del transporte en los países desarrollados, tanto en su infraestructura, operación, políticas, estructura interna, como en su filosofía. En Colombia el Transporte Público Colectivo Urbano es un servicio público en manos de particulares con fuertes intereses económicos. El Estado es el ente regulador, encargado de trazar las políticas del transporte público y de hacer cumplir las normas de tránsito. Los dueños de los vehículos los afilian a empresas de transporte y los conductores son contratados directamente por los dueños para trabajar por porcentaje. Así que es un sistema complejo en el cual intervienen muchos actores con intereses que pueden ser conflictivos entre sí. En la Figura 1 se observa una síntesis de la problemática actual del transporte público colectivo.

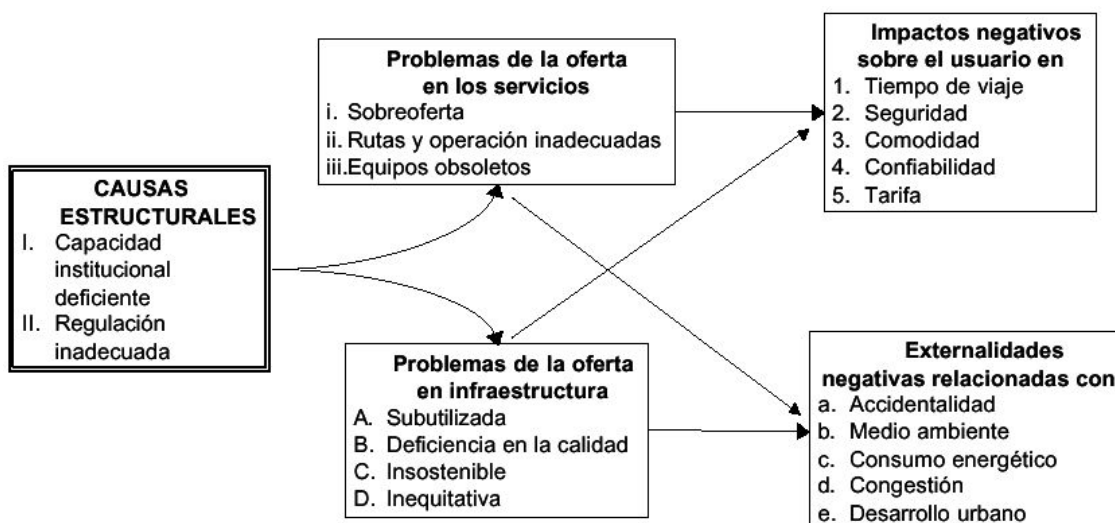


Figura 1. Problemática actual del Transporte.

Fuente: Documento CONPES 3260, Departamento Nacional de Planeación, 2003

El gobierno no ha tenido políticas claras con respecto al transporte. Ha sido débil ante los transportadores y ha permitido que los intereses particulares primen sobre el bienestar de la comunidad. El gobierno ejerce su función como ente regulador del transporte en cada una de las ciudades a través de la Secretaría de Tránsito y Transporte. Cada empresa de transporte público opera de manera independiente y sin mucho control por parte de la Secretaría de Tránsito. Además el Secretario de Tránsito es una figura política que cambia constantemente y que generalmente no tiene una fuerte formación técnica en lo relacionado con el Transporte Público.

Aunque la información y el análisis del problema, corresponde a las condiciones de operación de la ciudad de Manizales (una ciudad relativamente pequeña, situada en Colombia), los problemas y características de operación del transporte, son los mismos de las demás ciudades del país.

Los sistemas de transporte público en Colombia, son por lo general, sistemas que utilizan vehículos de baja capacidad. Muchos de los vehículos comparten los mismos corredores viales, no existen trasbordos, sino que todas las rutas cubren recorridos extensos. Ni siquiera hay coordinación entre las empresas que cubren las mismas rutas y no hay mucha tecnología para la operación y el control.

La mayoría de las empresas de Transporte Público Colectivo Urbano no son en realidad empresas transportadoras, ya que no son propietarias de los vehículos, sino más bien son afiliadoras o cooperativas que agrupan a una gran multitud de propietarios de vehículos que se afilian a una empresa en particular, para poder operar sus vehículos en las rutas de dicha empresa. Las rutas no corresponden a las necesidades de movilidad de los usuarios y existen rutas ilegales o no autorizadas por el gobierno. Además, las rutas autorizadas han sido modificadas de manera ilegal por las empresas (La Vial Ingenieros Ltda., 2001).

La empresa afiliadora simplemente cobra al dueño del vehículo una cuota única de afiliación denominada “cupó” y una cuota mensual de administración. Con esos dineros proporciona la infraestructura para el despacho y control de la operación en cada una de sus rutas. Sin embargo, dado que la empresa tiene garantizados sus ingresos por concepto de afiliación y administración, no se interesa mucho porque la

operación sea eficiente y sus mecanismos de control son demasiado ineficaces. De tal forma que para la empresa es mucho más importante aumentar la cantidad de vehículos afiliados que la movilización de pasajeros, la calidad del servicio y el cumplimiento de los itinerarios. Esa es la razón por la cual, con el aval del gobierno, han vinculado muchos más vehículos de los necesarios para atender la demanda y han generado una sobreoferta importante en el servicio, que ha generado muchos problemas, no solo para los usuarios, que deben pagar tarifas más altas, sino también para los propietarios y conductores, que han visto reducidos sus ingresos.

El propietario deriva sus ingresos del pago que los usuarios hacen por el servicio y por tal motivo su principal interés es movilizar el máximo número posible de pasajeros para lograr un nivel mayor de ganancias. Para completar el panorama, el conductor del vehículo tiene un sueldo que depende también de la movilización de pasajeros, lo cual lo obliga a pasar largas jornadas de trabajo luchando contra los demás conductores para tratar de movilizar el mayor número de pasajeros posible. Eso quiere decir que el Transporte Público Colectivo Urbano está organizado con un esquema de competencia desenfrenada que se ha denominado “guerra del centavo” (Moller, 2001), en la cual los conductores hacen cualquier cosa por movilizar más pasajeros, por ejemplo: incumplen los itinerarios, no respetan paraderos, conducen a velocidades peligrosas, compiten contra otros vehículos, se quedan esperando pasajeros y sobornan a los despachadores para obtener turnos de despacho mejores.

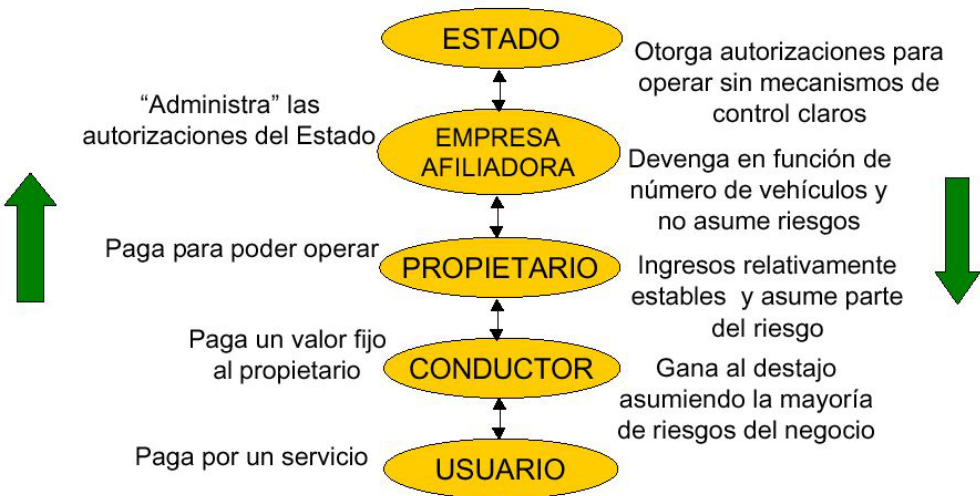


Figura 2. Esquema de funcionamiento actual.

Fuente: Documento CONPES 3260, Departamento Nacional de Planeación, 2003

En otros casos el propietario del vehículo cobra un dinero fijo diariamente al conductor, lo cual le garantiza unos ingresos más o menos estables, pero establece una presión mayor sobre el conductor que debe cubrir primero la cuota del propietario antes de empezar a ganar algo para sí mismo. El propietario del vehículo debe pagar la cuota de afiliación a la empresa y en la mayoría de los casos también tiene que cubrir una deuda a un banco por el préstamo que hizo para comprar el vehículo. Hoy en día, difícilmente le alcanza para cumplir con esas obligaciones y en ocasiones, termina vendiendo su vehículo porque ya no le permite sostener a su familia. Este fenómeno viene generando la desaparición de los pequeños propietarios y la aparición de propietarios con gran cantidad de vehículos, que ejercen su influencia para que sus carros tengan condiciones privilegiadas con respecto a los demás. Este esquema de funcionamiento se resume en la Figura 2.

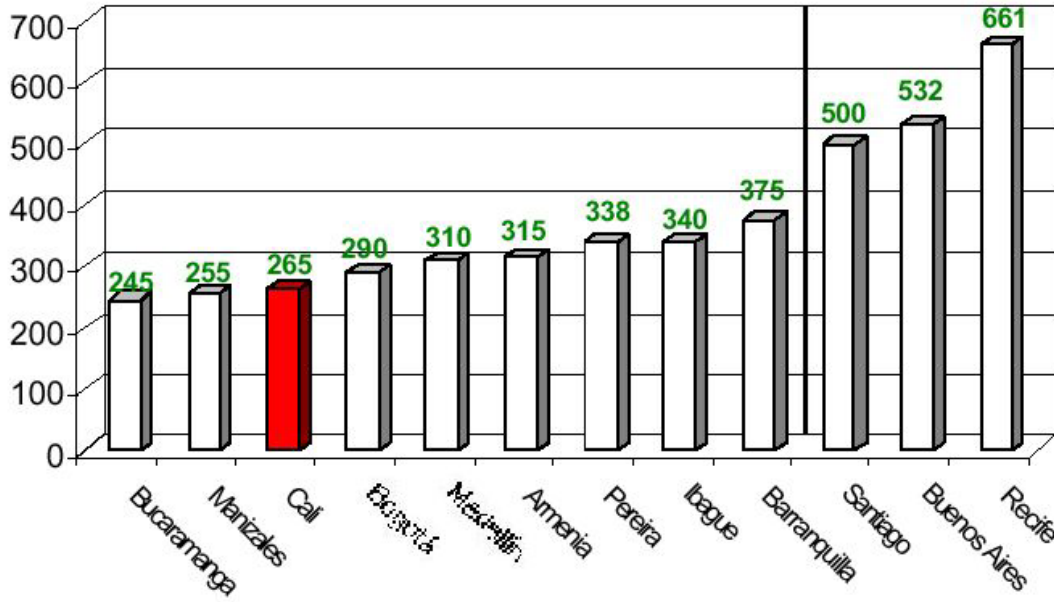


Figura 3. Pasajeros transportados por vehículo al día (2001).

Fuente: Política para mejorar el Transporte Público Urbano de pasajeros. DNP, 2002.

En Manizales, por ejemplo, hace algunos años los conductores trabajaban únicamente durante ciertos períodos de la jornada laboral y aún así movilizaban en promedio más de 1.000 pasajeros diarios. Hoy en día, debido a la sobreoferta de vehículos, la desorganización, las condiciones económicas y problemas como el desempleo, tienen que trabajar durante jornadas de más de 14 horas para obtener promedios de

movilización de pasajeros que no alcanzan a 300 pasajeros diarios. En la Figura 3 se observa un comparativo de la movilización de pasajeros por día en algunas ciudades del país con respecto a otras de Suramérica. Puede notarse que Manizales está en uno de los niveles más bajos de movilización y que todas las ciudades de Colombia están por debajo de las de Suramérica. Eso quiere decir que, dado que los ingresos se obtienen por concepto del pago que el usuario hace por el servicio, la baja movilización indica que el Transporte Público Colectivo Urbano ya no es un negocio tan lucrativo como lo fue en el pasado y que si no se toman medidas apropiadas para aumentar la eficiencia del servicio, muchos propietarios llegarán a la quiebra e inclusive algunas empresas de transporte público también.

“En la ciudad de Manizales existen seis (6) empresas autorizadas que prestan el servicio de Transporte Público Colectivo en la ciudad. Estas empresas tienen un total de 812 unidades, compuestas por buses, busetas y colectivos (se diferencian principalmente por el tamaño y la capacidad). Las busetas están clasificadas en servicio corriente, dorado y ejecutivo según la edad promedio de las mismas. Los buses y colectivos tienen un solo tipo de servicio” (La Vial Ingenieros Ltda., 2001).

La Tabla 1 contiene la relación de flota de las empresas, destacando la empresa Socobuses como la que posee el mayor número de vehículos, en conformidad con el mayor número de rutas que opera actualmente.

Tabla 1. Número de vehículos por empresa

Empresa		Gran	Servi-				
Servicio	Autolegal	Caldas	Turismo	Socobuses	Sideral	Unitrans	TOTAL
Buseta Ordinaria	5	10	0	82	2	23	122
Buseta Dorada	13	8	0	67	10	50	148
Buseta Ejecutiva	71	15	0	186	24	107	403
Bus	9	0	0	32	51	2	94
Colectivo	0	0	45	0	0	0	45
TOTAL	98	33	45	367	87	182	812

La cantidad de vehículos de transporte público que operan en la ciudad no corresponde a la demanda. Existen más vehículos de los necesarios para prestar un buen servicio en la ciudad, lo cual aumenta la congestión (especialmente en el centro),

reduce las ganancias de los propietarios y hace que el transporte público sea muy lento.

La medida gubernamental para contrarrestar la elevada sobreoferta de vehículos se conoce como “*pico y placa*” y consiste en la prohibición de la circulación de un determinado número de vehículos por día, para el caso de Manizales es el 10% del parque automotor. Los vehículos se dividen en grupos y a cada grupo le corresponde una letra; cada uno de los días de la semana (de lunes a sábado), descansan los vehículos correspondientes a una de esas letras. Sin embargo, esta medida no constituye una solución al Transporte Público Colectivo, ya que se limita a sacar de circulación un determinado número de vehículos por día, pero los que quedan en servicio siguen trabajando en completo desorden y sin una programación eficiente, ya que la totalidad de los vehículos que están en operación, circulan durante todo el día.

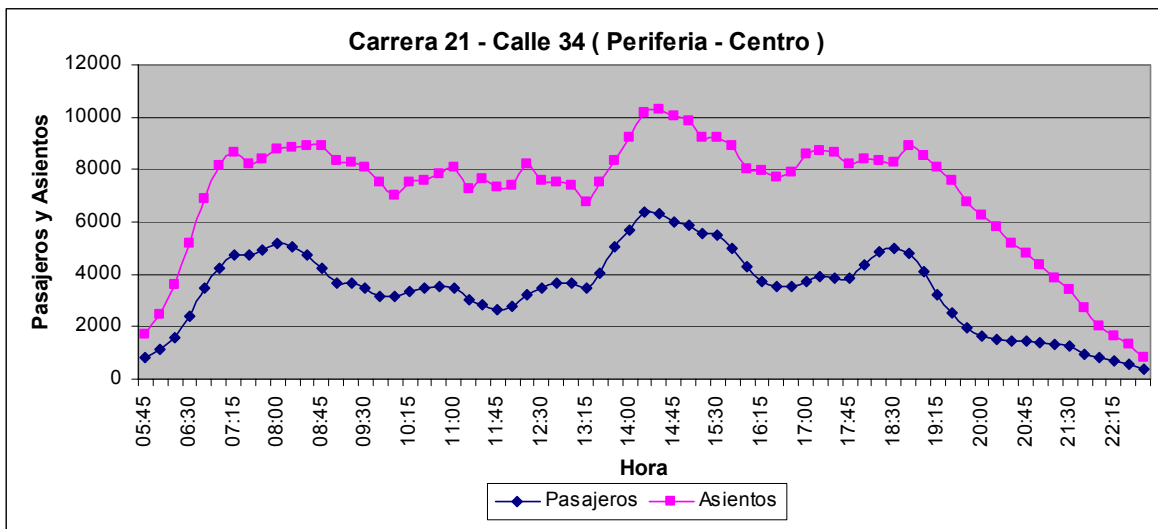


Figura 4. Oferta de asientos y demanda de pasajeros en el Centro de Manizales.

Fuente: La Vial Ingenieros Ltda. Actualización de las Variables de Transporte Público. Informe Final. Mayo de 2001.

La sobreoferta de sillas en puntos estratégicos de la red vial por donde pasan las rutas alcanza niveles altos. La sobreoferta estimada para el sistema es de aproximadamente el 38% (La Vial Ingenieros Ltda., 2001). En algunos sitios, para una hora pico, la sobreoferta alcanza más del 100%. La sobreoferta es aún mayor en las horas valle, debido a que hay menos pasajeros y circulan más vehículos de los necesarios. La figura 4 muestra la comparación de la oferta y demanda en una de las calles

principales del centro de la ciudad de Manizales. Puede observarse que siempre hay muchos más lugares que pasajeros.

Desde el punto de vista de infraestructura, también existen muchas deficiencias. Por ejemplo, no existen terminales de despacho con las condiciones mínimas para los vehículos y conductores. En la mayoría de los casos el Terminal de despacho es una tienda y los vehículos se parquean en la vía pública. En otros casos, existe un lote en el cual se parquean los vehículos pero el Terminal de despacho es una caseta que carece de las condiciones mínimas para la operación. En la figura 5 se observa el Terminal de despacho principal de la Empresa Socobuses S.A. (la más grande de la ciudad). A pesar de que este es uno de los mejores terminales de despacho, en la figura 6 se muestra la caseta del despachador, la cual carece de las condiciones mínimas de operación.



Figura 5. Principal Terminal de Despacho de la empresa Socobuses S.A.

Las deficiencias del Transporte Público Colectivo Urbano también se pueden ver desde diferentes puntos de vista:

Desde el punto de vista del pasajero “los buses son lentos porque paran constantemente para recoger y dejar pasajeros, se frenan mutuamente. Son a la vez causa y víctima de los trancones” (Melo, 2001). “Los buses no son calculables ni confiables. El pasajero no puede saber a qué hora va a pasar el bus, menos aún los domingos, días festivos y en la noche o muy temprano” (Moller, 2001). Los pasajeros están expuestos a excesiva contaminación y corren riesgos de accidentes subiendo y bajando en mitad de la calle y pueden sufrir golpes porque los conductores arrancan mientras el usuario sube o baja del vehículo.



Figura 6. Caseta de despacho del Terminal Principal de la empresa Socobuses S.A.

Desde el punto de vista del conductor, existen largas jornadas laborales de 12 y más horas, con efectos nocivos para la salud, los contactos sociales y la vida familiar del conductor. A pesar de eso los ingresos de los conductores son muy modestos. Las presiones para buscar pasajeros a toda costa resultan en altos riesgos de accidentes con otros vehículos.

Se podría llegar a pensar que la programación, operación y control ineficiente del Transporte Público, afecta exclusivamente a los actores que intervienen en el sistema, es decir, transportadores, conductores y usuarios, pero en realidad este problema afecta a negativamente a toda la comunidad e inclusive a toda la ciudad.

Los vehículos de transporte público contribuyen significativamente a la contaminación ambiental, pues generan alto consumo energético, alta emisión de partículas y altos niveles de ruido. De tal forma que la operación ineficiente del Transporte Público genera deterioro urbano y perjudica la calidad de vida en la ciudad. Para citar un ejemplo, los habitantes y comerciantes de un sector del centro de la ciudad de Manizales, interpusieron una Acción Popular contra las Empresas de Transporte para que reduzcan la cantidad de vehículos que circulan por ese sector, ya que tienen que soportar las inclemencias del ruido y el humo de los vehículos, que afecta las fachadas de los negocios, deteriora la mercancía y genera problemas respiratorios y auditivos

en los residentes. Estas personas realizaron un estudio que determinó que “...en un solo día pasan por una de las calles en conflicto (específicamente la Carrera 21) un total de 4.052 buses y busetas” (La Patria, 2004), una cifra escandalosa dado que es una vía de solo dos carriles. Además, menciona el estudio que los niveles de ruido generados sobrepasan el nivel permitido por el oído humano de 90 decibeles. “El aire que se inhala permanentemente tiene altas concentraciones de polución, partículas que paulatinamente van llegando a las vías aéreas terminales (alvéolos) y generan enfermedades pulmonares que llevarán a la obstrucción global de las vías aéreas y por ende a enfisemas y enfermedades pulmonares obstructivas crónicas”. (La Patria, 2004).

La congestión vehicular de las ciudades y la acumulación de vehículos que se detienen unos a otros, obedece principalmente a la deficiente programación de los vehículos de Transporte Público, que generan pérdida excesiva de tiempo para las personas haciéndolos soportar largos viajes que se reflejan en el cansancio y estrés que manifiestan al llegar a sus hogares o trabajos.

“La pérdida de tiempo en los viajes, el ruido, la contaminación y la agresividad en las relaciones entre diversos grupos en el tránsito, tienen como una de sus consecuencias el aumento del estrés. Ese estrés se manifiesta en relaciones sociales más agresivas al interior de las familias y en las empresas. Son más probables las relaciones tensas y peleas cuando los pasajeros vuelven a casa agotados después de un largo viaje en bus” (Moller, 2001).

“Se encontró que las empresas de transporte público no siguen un plan de operación que se ajuste a las variaciones horarias de la demanda de viajes” (La Vial Ingenieros Ltda., 2001). A lo largo de la jornada laboral el movimiento de pasajeros varía, por ejemplo hay horas de máxima demanda conocidas como “*horas pico*” en las que gran cantidad de usuarios se movilizan hacia o desde sus hogares, mientras hay otros intervalos en los que se reduce el movimiento de pasajeros y se conocen como “*horas valle*”. Esas variaciones exigen que se ajuste la frecuencia de despacho de los vehículos de Transporte Público, para que corresponda a las necesidades de movilización de los usuarios y esto implica que se va a necesitar un número diferente de vehículos para la operación en “*horas pico*” que para la operación en “*horas valle*”.

Actualmente no se tiene en cuenta esa variación y todos los vehículos operan durante toda la jornada laboral, generando una alta acumulación de vehículos en los terminales y disminuyendo el promedio de movilización de pasajeros durante las “horas valle”

Respecto al plan de operación, hay mucha variedad entre las diferentes empresas, pero ninguna de ellas tiene un programa de rodamientos eficiente, adecuado a la demanda y que sea respetado por parte de los despachadores y conductores. En el pasado los conductores trabajaban prácticamente como ellos querían, pero en la actualidad, tienen algún mecanismo de programación manual que busca que los vehículos cumplan con determinados despachos, por lo general, dentro de una misma ruta para todo un día.

Con respecto a la programación, casi todos los procesos son manuales. Pero esto no solo sucede con la programación, también sucede en la planeación, la operación y el control. Se podría decir que en este campo, no ha habido progresos significativos desde hace cincuenta años. Algunas empresas utilizan el computador para digitar la programación, pero todo el proceso sigue siendo manual.

A pesar de existir alguna programación, los despachadores la modifican según sus propios intereses y motivados por propinas que reciben de parte de los conductores. Además debido a que todo se maneja en papel, es muy difícil verificar el cumplimiento de la programación y obtener información confiable que facilite la planificación y el control. Para evidenciar esto se realizó un estudio de movilización de pasajeros en las principales rutas de transporte de la ciudad de Manizales, tomando datos de las horas de despacho, tiempos de recorrido y conteo de la registradora. En el estudio se encontró que no se respetan los horarios de despacho, algunos conductores hacen un par de vueltas y otros hacen hasta 10 vueltas, el movimiento de pasajeros es muy irregular al igual que el tiempo de recorrido. No se observan diferencias entre horas pico y horas valle. Los tiempos de espera son muy elevados y las jornadas de los conductores sobrepasan las 14 horas. Se utilizan muchos más vehículos de los necesarios para ofrecer un buen servicio.

En ninguno de los planes de rodamiento se sacan de operación vehículos durante las “horas valle”, lo cual genera esperas muy largas para todos, que alargan también la jornada laboral. Una programación eficiente no puede tener todos los vehículos operando durante todo el día, debe establecer períodos de tiempo en los que algunos vehículos salgan de operación para minimizar al máximo los tiempos de espera de los demás. También debería tener en cuenta la duración de la jornada laboral, el número de recorridos y un tiempo de descanso programado específicamente para tomar el almuerzo. Actualmente se tienen esperas muy largas después de cada uno de los despachos, especialmente en horas valle, pero no se tiene un descanso programado para el almuerzo.

El problema que se pretende resolver es la falta de optimización en la programación de los recorridos. Aunque la solución de este problema no acaba con toda la problemática del transporte, si puede representar muchas mejoras para todos los actores involucrados e incluso hacer más rentable y eficiente el servicio de transporte público colectivo, convirtiéndolo en una alternativa más atractiva para la comunidad. Hay otros problemas relacionados con el Transporte Público, cuya solución implica inversiones muy altas y cambios estructurales, que tal vez sean mucho más difíciles de implementar.

La implementación exitosa de un sistema de transporte masivo en la ciudad de Bogotá, denominado “*Transmilenio*”, ha impulsado a otras ciudades del país a desarrollar sistemas de transporte masivo. Sin embargo, el cubrimiento actual de este sistema es bajo y aún se requiere un sistema de transporte tradicional para los sectores que no tienen el sistema de transporte masivo y en dichos sectores se han profundizado los problemas del transporte. Esto se evidencia en un informe de la Revista Cambio sobre un estudio reciente realizado por el Centro de Estudios para el Desarrollo Económico (CEDE) acerca de los efectos del nuevo sistema de Transporte Público masivo implementado en la ciudad de Bogotá, el “*Transmilenio*” (Revista Cambio, 2004). El estudio mostró que: “El *Transmilenio* no ha disminuido trancones y accidentes, sólo los ha desplazado a otros sectores de la ciudad. A primera vista, las cifras parecerían apuntar a *Transmilenio* como la solución perfecta. Sólo dos años después de haber sido inaugurado, el viaje promedio en Bogotá se redujo de 44 minutos a 35, y el promedio de velocidad de los carros aumentó de 27 kilómetros por

hora a 32". El estudio también señala que "...en algunos sectores se respira un aire 9% más limpio y los accidentes de tránsito en las vías de Transmilenio cayeron 94%". Pero entonces, ¿por qué el balance negativo? El CEDE encontró que "...los malos hábitos del pasado podrían haber sido desplazados por *Transmilenio* a las rutas que éste no cubre. Actualmente, *Transmilenio* sólo moviliza al 13% de los usuarios de transporte público, pues son cerca de seis millones de personas las que aún dependen del sistema tradicional. En la puesta en marcha de Transmilenio sólo 1.410 buses fueron chatarrizados, mientras que otros 4.670 se desplazaron a otros sectores. Esto causó que el tiempo promedio de viaje en el resto del transporte público aumentara 10%".

Eso indica que mientras no se tenga un sistema de transporte masivo que cubra la totalidad de la ciudad de Bogotá, será necesario optimizar la programación de los vehículos del sistema tradicional para que la operación sea eficiente y no genere más problemas a la comunidad.

La programación del servicio se conoce como generación de rodamientos o itinerarios y el propósito es diseñar y construir un algoritmo de programación que facilite este proceso, generando programaciones que minimicen los tiempos de espera en Terminal, consideren descansos a lo largo de la jornada laboral, mantengan una asignación equitativa de recorridos entre los vehículos y que ofrezcan un servicio ajustado a la demanda. Además el algoritmo debe ser configurable para generar programaciones que atiendan a diferentes criterios y que se ajusten a las necesidades de cada una de las rutas y las empresas.

En otros países esa programación se efectúa mediante software especializado que generalmente incluye el diseño de las rutas. Sin embargo, estos paquetes de software están basados en modelos de optimización que no se ajustan a las condiciones de operación del Transporte Público en Colombia, por lo tanto se abordará el problema, analizando esos modelos de optimización y tratando de adaptarlos o en su defecto, generando un nuevo modelo acorde con las características y restricciones del transporte en el país.

A continuación se presentan las condiciones de operación y características de la programación, para mostrar porque se hace difícil adaptar los modelos convencionales a este sistema y para determinar los parámetros que se deben tener en cuenta en la optimización de la programación de rodamientos.

CAPITULO 3

PROGRAMACION DE RODAMIENTOS O ITINERARIOS PARA VEHICULOS DE TRANSPORTE PUBLICO COLECTIVO

La programación de rodamientos o itinerarios para los vehículos es un proceso que partiendo de las frecuencias de despacho para una determinada ruta a lo largo del día, determina los recorridos necesarios para prestar el servicio y asigna los vehículos que deben efectuar cada uno de esos recorridos. El objetivo es ofrecer un servicio ajustado a la demanda, pero organizando los despachos de cada vehículo de tal forma que se minimicen tiempos de espera en Terminal y se logre una programación lo más equitativa posible para los conductores. Este proceso se realiza para cada una de las rutas y abarca todos los días de la semana.

“Dentro de la operación del transporte tal vez no existe un aspecto más importante que las actividades de control y preparación de itinerarios. La comodidad y conveniencia del usuario, las compensaciones y las condiciones de trabajo de los operadores y las finanzas de la misma empresa están afectadas directa y significativamente por la programación de la operación. Por este motivo, se considera importante destacar la importancia de la elaboración de programas de servicio o itinerarios, así como el procedimiento detallado para su definición” (Molinero y Sánchez, 1998).

3.1 Importancia de los Rodamientos o Itinerarios.

La importancia de los rodamientos o itinerarios radica en que:

- Un rodamiento o itinerario establece la cantidad de servicio que se presta durante los diferentes períodos del día, conforme a la demanda esperada, permitiendo asegurar una oferta adecuada de servicio. En los casos donde los itinerarios son adecuados y los intervalos se mantienen, el usuario experimenta la conveniencia y comodidad de un buen servicio.
- Los rodamientos o itinerarios se relacionan con el tiempo que el usuario tiene que esperar en el paradero antes de abordar un vehículo y también se relacionan con el tiempo que se requiere para que un usuario se traslade desde un punto a otro dentro de la ruta. Por ello, la eficiencia de los rodamientos afectará al usuario en cuanto a la velocidad de los desplazamientos y el tiempo empleado en los viajes.
- Los rodamientos son importantes para los conductores, porque cuando se presentan deficiencias en la programación, las jornadas de trabajo son excesivamente largas y algunos conductores trabajan más que otros.
- Los rodamientos indican previamente al conductor la cantidad de trabajo que debe efectuar y las horas de inicio y terminación de cada recorrido.
- También son importantes para la empresa y los propietarios, puesto que son la base de una operación ordenada y controlada. La operación a intervalos inadecuados o irregulares, la operación a bajas velocidades y la asignación de un número inadecuado de unidades para satisfacer la demanda dan como resultado un servicio poco atractivo a los usuarios a la vez de redituar un menor ingreso al propietario y al conductor.
- Los itinerarios facilitan la supervisión del servicio al conocerse los tiempos en que los vehículos deben salir o llegar a un Terminal, pudiéndose observar las desviaciones en estos tiempos y facilitando la aplicación de medidas correctivas.
- El aspecto más significativo del proceso de programación de la operación se relaciona con los costos de operación, al estar éstos relacionados con la cantidad de servicio prestado. Una operación inadecuada y descuidada así como un rodamiento ineficiente, resulta en la utilización de un mayor parque vehicular, con el consecuente desperdicio del servicio y gasto de un mayor presupuesto de operación. Los rodamientos permiten absorber la máxima demanda al menor costo de operación.

3.2 Elementos Básicos de un rodamiento.

Antes de elaborar un rodamiento o itinerario es necesario definir los elementos básicos para su diseño. (Molinero y Sánchez, 1998) los define como sigue a continuación.

3.2.1 Horario de Servicio.

Se refiere a la hora de inicio del servicio en la mañana y la hora a la que se despacha el último vehículo en la noche. Entre estas dos horas se realiza toda la programación de despachos.

3.2.2 Intervalo.

Es la porción de tiempo, comúnmente expresada en minutos, entre dos salidas sucesivas de vehículos de Transporte Público en una ruta. El usuario está interesado en contar con un servicio con intervalos cortos para minimizar el tiempo de espera en los paraderos. Un intervalo corto implica para el transportador mayores costos de operación pero para el usuario representa un mejor servicio. Para determinar estos intervalos hay que tener en cuenta la necesidad de satisfacer la demanda y ofrecer una frecuencia mínima que mantenga un buen servicio.

3.2.3 Frecuencia de Servicio.

También se le denomina frecuencia de despacho y corresponde al número de vehículos que se despacha durante una hora (o cualquier período de tiempo considerado), siendo éste el inverso del intervalo.

3.2.4 Tiempo de Recorrido.

Es el intervalo de tiempo programado entre la salida de un Terminal y su llegada al Terminal opuesto en una ruta, o en el caso de rutas circulares, al mismo Terminal de partida.

3.2.5 Tiempo de Terminal

Es el tiempo adicional que un vehículo espera en un Terminal. Su propósito es contar con tiempo para dar vuelta al vehículo, para dar un descanso al conductor y para permitir los ajustes necesarios en el horario. Este tiempo permite además recuperar

las demoras en las que se ha incurrido en el despacho que acaba de terminar. Debe haber un tiempo mínimo de Terminal establecido por la empresa.

Para poder elaborar un rodamiento se requiere conocer para cada una de las rutas:

- El Horario de servicio.
- Los Intervalos y Frecuencias de servicio (varían de acuerdo a la hora del día y la demanda esperada).
- Tiempos de recorrido (También varían a lo largo del día, de acuerdo a las “*horas pico*” y “*horas valle*”).
- Tiempo de Terminal.

Los intervalos y frecuencias de servicio dependen de la movilización de pasajeros y la ocupación vehicular obtenida históricamente, además de la demanda estimada, que generalmente se obtienen a partir de Estudios de Tránsito. El tiempo de recorrido depende de las velocidades de operación y de la distancia que cubre la ruta.

La información anterior debe variar para diferentes períodos de tiempo, por ejemplo, es diferente para los domingos y festivos, días sábado y en semana. Además también varía a lo largo del año porque existen temporadas de estudio y vacaciones. Todo eso implica que deberían existir rodamientos o itinerarios diferentes para cada una de esas épocas, haciendo más complejo aún el proceso de programación.

3.3 Elaboración Manual de Rodamientos o Itinerarios (Método Clásico).

Las empresas que tienen programación de rodamientos, los elaboran de forma manual. El proceso manual de programación de rodamientos para una ruta es el siguiente: se tiene el tiempo de recorrido, se conocen los intervalos y frecuencias de servicio, que varían durante el día de acuerdo a las “*horas pico*” (horas de máxima demanda) y a las “*horas valle*” (horas de baja demanda). Con esa información se construye la tabla de despachos.

Por ejemplo: supóngase que se va a programar una ruta con un intervalo de despacho de 10 minutos entre las 7:00 a.m. y las 8:00 a.m., mientras que entre 8:00 a.m. y 10:00 a.m. el intervalo es de 20 minutos. El tiempo de recorrido es de 40 minutos. En la

programación manual que se realiza actualmente, se asigna previamente un determinado número de vehículos a la ruta, para este caso supongamos que son seis (6) vehículos. En la programación actual se busca programar esos seis (6) vehículos consecutivamente todo el día, lo cual genera una tabla de despacho como la que se observa en la Tabla 2. En dicha Tabla no se considera el Tiempo de Terminal, por lo tanto cada vehículo estará disponible para el siguiente despacho, sumando la hora de despacho y el tiempo de recorrido. Para el ejemplo, dado que la duración del recorrido es 40 minutos, el vehículo que se despacha a las 7:00 a.m. estará disponible para volverlo a programar a las 7:40 a.m., pero se despacha de nuevo a las 8:00 a.m. lo cual genera un tiempo de espera de 20 minutos antes de su segundo despacho. Después del segundo despacho, el vehículo queda disponible para volverse a despachar a las 8:40 a.m. pero se despacha a las 10:00 a.m. generando un tiempo de espera de 1 hora y 20 minutos.

Tabla 2. Ejemplo de Tabla de Despacho para la programación de 6 vehículos

Hora Despacho	Vehículo	Hora Disponible	Espera (min)
7:00 a.m.	1	7:40 a.m.	20:00
7:10 a.m.	2	7:50 a.m.	30:00
7:20 a.m.	3	8:00 a.m.	40:00
7:30 a.m.	4	8:10 a.m.	50:00
7:40 a.m.	5	8:20 a.m.	60:00
7:50 a.m.	6	8:30 a.m.	70:00
8:00 a.m.	1	8:40 a.m.	80:00
8:20 a.m.	2	9:00 a.m.	
8:40 a.m.	3	9:20 a.m.	
9:00 a.m.	4	9:40 a.m.	
9:20 a.m.	5	10:00 a.m.	
9:40 a.m.	6	10:20 a.m.	
10:00 a.m.	1	10:40 a.m.	
....

La ineficiencia de este modelo de programación se observa cuando se aumenta el intervalo de despacho, ya que se empiezan a incrementar los tiempos de espera de

los vehículos, que se quedan estorbando en el terminal o son despachados sin atender a la programación.

Entre los principales objetivos de la optimización están: disminuir el número de vehículos necesarios para operar la ruta y minimizar los tiempos de espera en terminal. Pero para lograr dicho objetivo no se pueden utilizar todos los vehículos durante todo el día, se deben programar descansos para los vehículos en las horas de menor demanda con el fin de reducir los tiempos de espera de los demás. En la actualidad no hay necesidad de programar tiempos de descanso para almorzar, debido a que hay tanto tiempo de espera acumulado durante el día que en uno de esos descansos se puede almorzar. La programación óptima no tendrá tiempos de espera prolongados, por lo cual será necesario programar el momento y la duración del almuerzo para cada uno de los vehículos.

Tabla 3. Ejemplo de Tabla de Despachos sacando vehículos a descansar

Hora Despacho	Vehículo	Hora Disponible	Espera (min)
7:00 a.m.	1	7:40 a.m.	
7:10 a.m.	2	7:50 a.m.	
7:20 a.m.	3	8:00 a.m.	0:00
7:30 a.m.	4	8:10 a.m.	10:00
7:40 a.m.	5	8:20 a.m.	
7:50 a.m.	6	8:30 a.m.	
8:00 a.m.	3	8:40 a.m.	0:00
8:20 a.m.	4	9:00 a.m.	0:00
8:40 a.m.	3	9:20 a.m.	0:00
9:00 a.m.	4	9:40 a.m.	0:00
9:20 a.m.	3	10:00 a.m.	0:00
9:40 a.m.	4	10:20 a.m.	
10:00 a.m.	3	10:40 a.m.	
....

Para entender mejor lo que se puede lograr haciendo ajustes al modelo de despacho, se presenta en la Tabla 3 una posible optimización de la programación mostrada en la

Tabla 2. Lo que se hizo fue en lugar de incluir todos los vehículos consecutivamente, se sacaron a descansar después del primer despacho los vehículos 1, 2, 5 y 6. Esto reduce los tiempos de espera de manera dramática, pero el problema es que en este fragmento de programación los vehículos 1, 2, 5 y 6 tuvieron 1 solo despacho, mientras que los vehículos 3 y 4 fueron programados 5 y 4 veces, respectivamente. Aquí se obtendría una mejora en cuanto a minimizar tiempos de espera, pero no es una programación óptima en el sentido de que no hay equidad para los vehículos (unos tuvieron 1 solo despacho y otros más de 3). La optimización que se pretende desarrollar debe minimizar los tiempos de espera pero buscando que todos los vehículos tengan aproximadamente el mismo número de despachos durante el día y programando eficientemente los almuerzos. Esto obedece a que en el modelo del Transporte Público Urbano en Colombia, los ingresos de los propietarios y conductores dependen de la movilización de pasajeros y ésta puede ser mayor en la medida que realicen mayor número de recorridos en el día. Adicionalmente, existe una restricción respecto al número máximo de despachos que puede tener un vehículo antes de tener un descanso, esto con el fin de garantizar unas condiciones de trabajo apropiadas para los conductores.

El problema comienza a complicarse, debido a la cantidad de restricciones que se deben incluir para lograr una programación óptima.

3.4 Elaboración del Rodamiento o Itinerario para la ruta piloto.

A continuación se explicará el proceso de programación manual para una ruta de la Empresa Autolegal S.A., denominada Ruta Estambul-Universidad-Estambul. Es una ruta circular cuyo Terminal de despacho y de llegada es Estambul. La empresa suministró los datos necesarios para la programación de los rodamientos. Estos datos corresponden a la programación para temporada de estudio en días hábiles (lunes a viernes). Los parámetros de operación, entregados por la empresa, necesarios para la elaboración de los rodamientos son los siguientes:

- ✓ Horario de Servicio: Desde las 5:40 a.m. hasta las 9:10 p.m.
- ✓ Tiempo de terminal: 5 minutos.
- ✓ Los intervalos de despacho y tiempos de recorrido suministrados por la empresa aparecen en la Tabla 4.

Se escogió esta ruta como piloto para las pruebas, debido a que puede ser una ruta compleja de programar por las variaciones de frecuencia que se presentan a lo largo del día, que corresponden a las “horas pico” y “horas valle”. Estas variaciones de frecuencia y tiempos de recorrido a lo largo del día hacen necesarios una cantidad diferente de vehículos, es decir, que no todos los vehículos asignados a la ruta deberían estar circulando durante todo el día.

Tabla 4. Intervalos de despacho para la ruta piloto

Desde	Hasta	Frecuencia de despacho	Tiempo de Recorrido
5:40 a.m.	6:09 a.m.	8 minutos	1:26 minutos
6:10 a.m.	7:39 a.m.	5 minutos	1:30 minutos
7:40 a.m.	10:59 a.m.	7 minutos	1:28 minutos
11:00 a.m.	11:59 a.m.	5 minutos	1:30 minutos
12:00 p.m.	12:54 p.m.	7 minutos	1:28 minutos
12:55 p.m.	1:59 p.m.	5 minutos	1:30 minutos
2:00 p.m.	4:59 p.m.	7 minutos	1:28 minutos
5:00 p.m.	6:39 p.m.	5 minutos	1:30 minutos
6:40 p.m.	7:50 p.m.	7 minutos	1:28 minutos
7:50 p.m.	8:29 p.m.	10 minutos	1:26 minutos
8:30 p.m.	9:10 p.m.	20 minutos	1:26 minutos

Para generar la tabla de despachos, se utiliza la información sobre los intervalos (véase Tabla 4). Con esa información se puede calcular cada cuanto tiempo debe efectuarse un despacho y adicionalmente se calcula la hora a la que estará disponible el vehículo que sea asignado, tomando la hora de despacho y sumándole el tiempo de recorrido y el tiempo mínimo de descanso entre recorridos. Por ejemplo, para el primer despacho que es a las 5:40 a.m., el vehículo que se despache a esa hora estará disponible para volver a ser despachado a las 7:11 a.m., que es el resultado de sumarle a las 5:40 a.m., 1 hora y 26 minutos de recorrido más 5 minutos de descanso. Con esa información se genera la Tabla 5, dejando una columna en la cual se anotarán los vehículos o rutinas correspondientes. Para la ruta en cuestión serían necesarios 147 despachos diarios, eso quiere decir que la tabla de despachos debería tener 147 filas, pero no se incluyeron todas en la tabla 5, porque es solo una tabla

ilustrativa. En el Capítulo 6, se utilizará la tabla de despachos completa para la programación.

Si la Tabla 5 se llena con información específica de vehículos (por ejemplo el número interno), se estaría programando un solo día de servicio. Entonces para hacer más eficiente la programación, se utiliza el término rutina, que quiere decir, la asignación completa para un vehículo durante un día. De esta forma la tabla se llena con números de rutinas y posteriormente se asigna a cada uno de los vehículos las rutinas que deben cumplir a lo largo de una semana o un período de tiempo mayor. Entonces se utiliza la misma tabla para lunes, martes, miércoles, jueves y viernes, solo que, por ejemplo, la rutina 7 será asignada el lunes al vehículo 44, el martes al vehículo 27, el miércoles al vehículo 12, etc.

Tabla 5. Tabla de despachos para un día

Hora Despacho	Rutina	Hora Disponible
5:40:00 a.m.		7:11:00 a.m.
5:48:00 a.m.		7:19:00 a.m.
5:56:00 a.m.		7:27:00 a.m.
6:04:00 a.m.		7:35:00 a.m.
6:10:00 a.m.		7:45:00 a.m.
6:15:00 a.m.		7:50:00 a.m.
6:20:00 a.m.		7:55:00 a.m.
6:25:00 a.m.		8:00:00 a.m.
6:30:00 a.m.		8:05:00 a.m.
6:35:00 a.m.		8:10:00 a.m.
6:40:00 a.m.		8:15:00 a.m.
6:45:00 a.m.		8:20:00 a.m.
6:50:00 a.m.		8:25:00 a.m.
6:55:00 a.m.		8:30:00 a.m.
7:00:00 a.m.		8:35:00 a.m.
...
...
8:20:00 p.m.		9:51:00 p.m.
8:30:00 p.m.		10:01:00 p.m.
8:50:00 p.m.		10:21:00 p.m.
9:10:00 p.m.		10:41:00 p.m.

Una vez se obtenga la programación óptima para un día, se hará la asignación de rutinas para períodos de tiempo más largos, tales como semanas o meses, utilizando programación lineal entera.

Cuando se ha llenado completamente la Tabla 5. Se puede extraer un informe de los despachos o recorridos que debe cubrir cada una de las rutinas a lo largo del día. Ese listado es el que se le entrega a cada uno de los conductores, ya que contiene todos los despachos que debe efectuar en un día de trabajo. Como ejemplo, se incluye la programación de una rutina en la Tabla 6.

Tabla 6. Programación de una Rutina

Hora Despacho	Rutina	Vuelta	Hora Disponible	Espera
5:56:00 a.m.	0	1	7:27:00 a.m.	0:03:00
7:30:00 a.m.	0	2	9:05:00 a.m.	2:00:00
11:05:00 a.m.	0	3	12:40:00 p.m.	0:02:00
12:42:00 p.m.	0	4	2:15:00 p.m.	0:06:00
2:21:00 p.m.	0	5	3:54:00 p.m.	0:05:00
3:59:00 p.m.	0	6	5:32:00 p.m.	0:03:00
5:35:00 p.m.	0	7	7:10:00 p.m.	

Si se analiza con detalle la Tabla 6 se pueden observar detalles importantes, como las horas a las que va a ser despachado el vehículo que sea asignado a la rutina 0. Esta rutina tendrá en el día un total de 7 despachos, que se denominan en la tabla “vuelta”. La columna “*Hora Disponible*” se obtiene sumando a la hora de despacho el tiempo de recorrido y el tiempo de Terminal. Restando la hora del siguiente despacho de la hora a la que estaba disponible, se puede calcular la columna “espera”, que corresponde al tiempo que el vehículo pasa en el Terminal antes de que se ponga en servicio de nuevo. Es importante aclarar que ese tiempo de descanso es adicional a los 5 minutos de Tiempo de Terminal, que están incluidos en el cálculo de la hora disponible. Por ejemplo, el segundo despacho es a las 7:30 a.m. y el vehículo estará disponible a las 7:27 a.m., lo que indica que cuando se despacha el vehículo a las 7:30 a.m. ha tenido un descanso adicional de 3 minutos. Uno de los objetivos de la

programación eficiente es reducir esos períodos de descanso al mínimo, sin embargo, también es necesario programar un tiempo de descanso más largo que permita al conductor almorzar. En este ejemplo, el conductor podría almorzar entre las 9:05 a.m. y las 11:05 a.m. (2 horas). Sería mucho mejor que ese almuerzo fuera más tarde, por ejemplo, entre la 1:00 p.m. y las 3:00 p.m., pero eso no siempre es posible para todos los vehículos.

Aunque la mayoría de las empresas realizan la programación de la misma manera, existen diferencias importantes. Por ejemplo, algunas empresas programan sus vehículos para que operen durante todo el día en una sola ruta, mientras que otras empresas los rotan entre diferentes rutas a lo largo del día. Sin embargo, el método manual, aplica en general a todas las empresas.

En este capítulo se ha explicado la importancia de la programación de rodamientos y lo que se requiere para poder elaborarlos. Se han presentado las generalidades del método manual de programación para una ruta piloto, pero en el Capítulo 6 se efectuará la programación completa para esa ruta utilizando varios métodos y se compararán los resultados.

En el próximo capítulo se explicarán los fundamentos de una técnica de optimización conocida como algoritmos genéticos y en el capítulo 5 se diseñará un algoritmo genético especial para tratar de optimizar el modelo de programación de rodamientos.

CAPITULO 4

INTRODUCCION A LOS ALGORITMOS GENETICOS

Para la programación óptima de itinerarios o rodamientos no se utilizaron las técnicas clásicas por varias razones:

- Es complejo establecer el modelo matemático.
- El problema presenta alta dimensionalidad.
- Algunas de las funciones objetivo no son lineales. Algunas son imposibles de expresar como una función y otras solo se pueden expresar como funciones a tramos.

En este capítulo se explican los fundamentos de los algoritmos genéticos (AG) y se destacan sus ventajas comparados con las técnicas convencionales y con otras metaheurísticas.

Los algoritmos genéticos son un mecanismo de optimización que simula el proceso de evolución por selección natural y están basados en la genética. En Ross et al. (1994) se definen como "...una estrategia de búsqueda inusual". Pero dado que se basan en conceptos biológicos, a continuación se explican algunos de esos conceptos.

Cada individuo posee ciertas características que lo identifican. En el caso de los seres humanos, cada uno posee cierta estatura, cierto color de ojos y cierto tipo sanguíneo. Estas características "externas", constituyen lo que se denomina el fenotipo de un individuo. Cada una de estas características es resultado de la interacción del medio ambiente en que se desarrolla un individuo y la herencia que éste recibe de sus ancestros.

El fenotipo de cada individuo está definido en la información genética contenida en sus cromosomas. En cada célula existen dos juegos de cromosomas que definen las mismas características; un juego es aportado por el padre del individuo y el otro por la madre.

Los cromosomas están formados por genes y el valor que posee un gen determinado se denomina alelo.

Hay unas células especiales llamadas gametos, que intervienen en la reproducción (los espermatozoides y los óvulos humanos pertenecen a esta categoría). Los gametos se reproducen por meiosis, un proceso que consiste en:

1. Se duplica el número de cromosomas en la célula, esto es, se hace una copia de cada cromosoma. Al final quedan dos juegos correspondientes al padre y dos a la madre. Hay una enzima encargada de copiarlos, ésta es la ADN-polimerasa. La molécula de ADN tiene forma de una doble hélice, como una escalera de caracol. La enzima abre por en medio los “escalones” de esta hélice y ensambla en cada mitad los nucleótidos que debe ensamblar. Ocasionalmente esta enzima comete un error, que puede ser causado por radiaciones energéticas externas o sustancias extrañas. La alteración de la molécula de ADN original constituye una mutación que puede manifestarse en el fenotipo y hacer al individuo diferente del resto de sus congéneres. Por lo general las mutaciones son desfavorables, incluso letales, para el organismo mutante. Pero ocasionalmente pueden no serlo y conferirle a dicho organismo alguna ventaja que le permita sobrevivir más fácilmente en su medio. Esta característica será transmitida a sus descendientes y un pequeño paso evolutivo se habrá dado.

2. Se cruzan un juego de cromosomas del padre con uno de la madre, formándose dos juegos de cromosomas híbridos. El resultado es un juego de cromosomas puros del padre, un juego puro de la madre y dos juegos de cromosomas híbridos. Aquí se mezclan las características del padre y la madre. Para el cruzamiento de dos cromosomas se forman entre ellos puntos de ruptura y unión de las cadenas de ADN. Estos puntos, llamados quiasmas, cortan el cromosoma en segmentos llamados cromátidas y unen cromátidas complementarias de dos distintos cromosomas. Al final

cada cromosoma que participó en la cruce queda constituido por segmentos que ya poseía y por otros que eran de su análogo.

Los algoritmos genéticos tratan de simular el mismo proceso. Es necesario codificar de alguna manera el dominio del problema para obtener estructuras manejables que puedan ser manipuladas por el Algoritmo Genético. Cada una de estas estructuras constituye el equivalente al genotipo de un individuo en términos biológicos.

Es frecuente que el código de los elementos del dominio del problema utilice un alfabeto binario (0's y 1's). Una vez que se ha definido la manera de codificar los elementos del dominio del problema y se conoce la forma de pasar de un elemento a su código y viceversa, es necesario fijar un punto de partida. Los algoritmos genéticos manipulan conjuntos de códigos en generaciones sucesivas o poblaciones, en ellas un código puede aparecer más de una vez. El algoritmo se encargará de favorecer la aparición en la población de códigos que correspondan a elementos del dominio que estén próximos a resolver el problema. En resumen, el algoritmo recibirá como entrada una población de códigos y a partir de ésta generará nuevas poblaciones, donde algunos códigos desaparecerán mientras que otros, que se mapean en mejores soluciones posibles, aparecen con más frecuencia hasta que se encuentra una solución satisfactoria o hasta que se cumple alguna otra condición de terminación. A los códigos en una población, es decir, los elementos de ésta se les llama individuos.

4.1 Evaluación de la población

En la naturaleza hay individuos más hábiles que otros para sobrevivir. Al igual que en la naturaleza, en los algoritmos genéticos es necesario establecer algún criterio que permita decidir cuáles de las soluciones propuestas en una población son mejores respecto del resto de las propuestas y cuáles no lo son. Es necesario establecer, para cada individuo, una medida de desempeño relativa a la población a la que pertenece. Cada individuo de cada generación de un algoritmo genético recibe una calificación o una medida de su grado de adaptación (fitness). El objetivo de este número es que permita distinguir propuestas de solución buenas de aquéllas que no lo son.

4.2 Mecanismos de Selección

Una vez calificados todos los individuos de una generación, el algoritmo debe seleccionar a los individuos más calificados, mejor adaptados al medio, para que tengan mayor oportunidad de reproducción. De esta forma se incrementa la probabilidad de tener individuos “buenos” en el futuro. La selección ocasiona que haya más individuos buenos, explota el conocimiento que se ha obtenido hasta el momento, procurando elegir lo mejor que se haya encontrado, elevando así el nivel de adaptación de toda la población.

En principio podría parecer que es conveniente tener una estrategia de selección estricta para que mejore rápidamente la población y converja el algoritmo, es decir, que la población se acumule alrededor de un genotipo óptimo. Esto no es cierto. Lo que ocurrirá es que la población se acumulará rápidamente alrededor de algún individuo que sea bueno, comparativamente con el resto de los individuos considerados a lo largo de la ejecución del algoritmo, pero este individuo puede no ser el mejor posible. A esto se le suele llamar convergencia prematura.

No se puede asegurar pero sí procurar que lo anterior no ocurra. Además de la explotación es necesario que exista exploración. El AG debe, no sólo seleccionar de entre lo mejor que ha encontrado, sino procurar encontrar mejores individuos.

En la estrategia de selección normalmente hace que sea más probable elegir al mejor individuo de la población pero sin asegurarlo, es posible que este individuo se pierda y no forme parte de la siguiente generación. Para evitar lo anterior se puede obligar que los mejores individuos de una generación pasen intactos a la siguiente. A esta estrategia se le denomina elitismo.

Un método de selección de padres muy usado es el de selección por torneo binario. En este, cada cadena de la población inicial tiene dos copias en la población del torneo. Esta población se entremezcla para obtener pares aleatorios que competirán entre ellos y el mejor sobrevive. Las ganadoras ganaran dos torneos y se presentarán dos veces en la selección. Este método puede ser desventajoso porque puede hacer perder diversidad en la búsqueda.

4.3 Cruzamiento

La cruce de los códigos genéticos de individuos exitosos favorece la aparición de nuevos individuos que hereden de sus ancestros características deseables. Dados dos individuos seleccionados en función de su grado de adaptación, éstos pasan a formar parte de la siguiente generación o, al menos, mezclan sus códigos genéticos para generar hijos que posean un código híbrido. Existen muchos mecanismos de cruzamiento.

El mecanismo de cruzamiento más común es el llamado cruzamiento de un punto. En este tipo de cruzamiento, dados dos individuos se elige aleatoriamente un punto de corte entre dos bits cualesquiera del cromosoma. Esto define segmentos izquierdos y derechos en cada genotipo. Se procede entonces a intercambiar los segmentos derechos (o los izquierdos, indistintamente) de cada individuo.

4.4 Mutación

Ocasionalmente algunos elementos del código de ciertos individuos de un algoritmo genético se alteran a propósito. Éstos se seleccionan aleatoriamente en lo que constituye el símil de una mutación. El objetivo es generar nuevos individuos, que exploren regiones del dominio del problema que probablemente no se han visitado aún. Esta exploración no presupone conocimiento alguno, no es sesgada.

4.5 El algoritmo genético simple (AGS)

Hay varios tipos de algoritmos genéticos. En Goldberg (1989) se denomina algoritmo genético simple (AGS) a uno que tiene las siguientes características:

- Tamaño de población fijo en todas las generaciones.
- Selección proporcional (de ruleta).
- Cruzamiento de un punto. La probabilidad de cruzamiento se mantiene fija para todas las generaciones y todas las parejas.

- Mutación uniforme (todas las posiciones de las cadenas genéticas tienen la misma probabilidad de ser cambiadas). La probabilidad de mutación permanece fija para todas las generaciones y todas las posiciones de los individuos.
- Selección no elitista. Esto es, no se copian individuos de una generación a otra sin pasar por el proceso de selección aleatoria (en este caso, proporcional).

El algoritmo es como sigue:

1. Codificar el dominio del problema en binario.
2. Generar un conjunto aleatorio (población inicial) de N posibles soluciones codificadas al problema. A ésta se le llamará la población actual.
3. Calificar cada posible solución (individuo) de la población actual.
4. Seleccionar dos individuos de la población actual con una probabilidad proporcional a su calificación.
5. generar un número aleatorio.
6. Si el número aleatorio es mayor que la probabilidad de cruzamiento se deben mezclar los códigos de los dos individuos seleccionados (cruzamiento de un punto) para formar dos híbridos, a los que llamaremos nuevos individuos.
7. Si el número aleatorio es menor que la probabilidad de cruzamiento llamamos a los individuos seleccionados nuevos individuos.
8. Por cada bit de cada nuevo individuo se genera un número aleatorio.
9. Si el número aleatorio es menor que la probabilidad de mutación se debe cambiar el bit en turno por su complemento.
10. Si el número aleatorio es mayor que la probabilidad de mutación el bit permanece inalterado.
11. Incluir a los dos nuevos individuos en una nueva población.
12. Si la nueva población tiene ya N individuos, llamarla población actual y regresar al paso 3, a menos que se cumpla alguna condición de terminación.
13. Si no, regresar al paso 4.

Para la optimización de la programación de rodamientos en vehículos de transporte público, se parte de este algoritmo básico pero son necesarias modificaciones sustanciales para que se adapte a las características y restricciones del problema en

cuestión. Por ejemplo, para el algoritmo genético que se diseñó, se utilizó codificación decimal y no binaria.

4.6 Funciones de penalización

El uso de penalizaciones respecto a las restricciones ha sido muy bien aceptado en investigaciones posteriores al inicio del uso de los algoritmos genéticos. La forma de penalizar las cadenas no factibles no es de ningún modo trivial ni existe una regla general de penalización. En problemas en los que el espacio de decisión es muy reducido debido a gran número de restricciones puede ser efectivo. Debido a que en Investigación de Operaciones los problemas suelen ser de este tipo, resulta muy interesante su investigación dentro de ésta área

Aunque son pocos los estudios realizados al respecto, puede destacarse algunas propuestas de funciones penalizadoras para las cadenas no factibles \mathbf{x}_k . Todas ellas se basan en la violación de cada restricción j , g_j^p , tal que:

$$(4.1) \quad g_j^p(\mathbf{x}_k) = \max\{0, g_j(\mathbf{x}_k)\} \quad , \text{ si la restricción es de tipo } g_j \leq 0$$

$$g_j^p(\mathbf{x}_k) = |g_j(\mathbf{x}_k)| \quad , \text{ si la restricción es de tipo } g_j = 0$$

Algunos de los métodos destacables son:

- Método de Homaifar, Lai y Qi:

Asume que para cada restricción se establece una familia de intervalos que determinan valores de penalización apropiados, es decir, para cada restricción el usuario crea algunos niveles de violación. Para cada nivel de violación i y para cada restricción j crea además un coeficiente de penalización R_{ij} . Si se está minimizando Z , evalúa luego las cadenas no factibles usando:

$$(4.2) \quad Z^p(\mathbf{x}_k) = Z(\mathbf{x}_k) + \sum_{i=1}^m R_{ij} g_j^p{}^2(\mathbf{x}_k)$$

$Z(\mathbf{x}_k)$ es la función de ajuste o función objetivo de la cadena \mathbf{x}_k y $Z^p(\mathbf{x}_k)$ es el valor de $Z(\mathbf{x}_k)$ penalizado para compararlo con las demás cadenas factibles o no factibles.

Este método presenta la desventaja de que requiere de muchos parámetros. Si estos son moderados podrían resaltar dentro de la población cadenas muy poco factibles y si son muy altos puede ser semejante al algoritmo que rechaza todas las cadenas no factibles, perdiendo diversidad en la población.

– Método de Joines y Houck

Este método asume que las penalizaciones a una cadena no factible dependen de la iteración presente. Si se está minimizando Z , en una iteración t es evaluada de la siguiente manera:

$$(4.3) \quad Z^p(\mathbf{x}_k) = Z(\mathbf{x}_k) + (C.t)^\alpha \sum_{i=1}^m g^p_j{}^\beta(\mathbf{x}_k)$$

A diferencia del método anterior, se requiere de menos parámetros. Los parámetros C , α y β no dependen del número de restricciones.

– Método de Powell y Skolnick

Incorporan una regla heurística para penalizar las cadenas no factibles, basada en que ninguna solución no factible puede tener un valor mejor en la función de ajuste que cualquiera de las cadenas factibles de la iteración. Ellos proponen:

$$(4.4) \quad Z^p(\mathbf{x}_k) = Z(\mathbf{x}_k) + r \sum_{i=1}^m g^p_j(\mathbf{x}_k) + \rho(\mathbf{x}, t)$$

donde r es una constante y ρ es de la forma:

$$(4.5) \quad \rho(\mathbf{x}, t) = \max\{0, \max\{Z(\mathbf{x})\} - \min\{Z^p(\mathbf{x})\}\}$$

donde $\max\{Z(\mathbf{x})\}$ es el máximo de los valores de las funciones objetivo de las cadenas factibles y $\min\{Z^p(\mathbf{x})\}$ es el mínimo de los valores de las funciones objetivo de las cadenas no factibles.

La ventaja de este método es que cadenas no factibles, con muy buenos valores originales en la función objetivo no se destacan notablemente dentro de la población.

- Métodos basados en optimización multiobjetivo

Estos métodos utilizan la teoría básica de la optimización ponderada de varios objetivos. De esta forma, la evaluación de \mathbf{x}_k puede hacerse de la forma:

$$(4.6) \quad Z^p(\mathbf{x}_k) = Z(\mathbf{x}_k) + \sum_{i=1}^m \omega_j g^p_j(\mathbf{x}_k)$$

donde los pesos ω_j reflejan la importancia relativa que se le da a cada restricción. Este método puede requerir de muchos parámetros si se tienen muchas restricciones.

En general, todas las funciones penalizadoras que se han descrito son útiles dependiendo del problema. La elección de una función de penalización depende del tamaño del espacio factible respecto al espacio de búsqueda completo, las propiedades topológicas del espacio factible, el tipo de la función objetivo, el número de variables, el número y tipo de las restricciones y el número de restricciones activas en el óptimo.

4.7 Optimización de múltiples objetivos

El análisis de problemas decisionales con criterios múltiples constituye quizás el área de desarrollo más activo en los últimos años en el campo de las ciencias de decisión (investigación operativa, gestión de recursos, etc.) (Romero, 1993).

En muchos problemas de decisión se considera alcanzar un conjunto de múltiples objetivos. Tal problema se acostumbra expresar matemáticamente de la siguiente manera general:

$$(4.7) \quad \text{Maximizar } \mathbf{Z}(\mathbf{x}) = (Z_1(\mathbf{x}), Z_2(\mathbf{x}), \dots, Z_q(\mathbf{x}))$$

sujeto a:

$$(4.8) \quad g_k(\mathbf{x}) \leq 0, \quad k = 1, \dots, m$$

Donde \mathbf{Z} es el vector de las q funciones objetivo Z_i , \mathbf{x} es el vector de las n variables de decisión y las funciones g_k son m restricciones que definen el espacio factible \mathbf{S} . Las restricciones, con frecuencia, están ligadas a la cantidad de recursos y capital

disponibles. No existe una solución factible x que maximice simultáneamente todos los objetivos. Debe buscarse entonces una solución satisfactoria y eficiente o no dominada, que aproveche al máximo los recursos, es decir, si se desea mejorar en algún objetivo, esto sólo podría hacerse a expensas de otro. Si el espacio de decisión es continuo, existen infinitas soluciones eficientes o no dominadas (una solución domina a otra cuando es mejor respecto a todos los criterios en consideración). Es mediante el juicio del decisor que puede escogerse una de ellas como solución al problema. Por lo tanto su resolución requiere incorporar valoraciones objetivas y subjetivas, esta última expresadas mediante la estructura de preferencias del decisor.

Las metodologías de análisis multiobjetivo presentan muy diversas maneras de obtener y representar numéricamente la estructura de preferencias del decisor. Las más usadas son los pesos de importancia relativa. Ellos son usados en los factores ponderantes (Zadeh, 1963), programación por compromiso (Zeleny, 1973), programación por metas (Charnes y Cooper, 1961), ELECTRE I, II y IV (Roy, 1968), función de utilidad multicriterio (Goicochea y otros, 1982), entre otros. (Detalles de todas las metodologías se encuentran en Goicochea et al, 1982 y Smith et al, 2000).

Para la optimización de varios objetivos, existen cuatro métodos basados en algoritmos genéticos:

- Suma ponderada de los diferentes objetivos:

Se optimiza una función que pondera linealmente las funciones objetivo. Un problema asociado a este método es que la curva estimada no garantiza que encuentre puntos no convexos. Solo ofrecerá además una solución por proceso genético realizado.

- Método VEGA:

En cada etapa de selección se escoge aleatoriamente (con una probabilidad específica), uno de los objetivos para determinar la bondad de la cadena solo con el valor de ese objetivo. Por ejemplo para dos objetivos $P_{\text{elección}}$ es la probabilidad de que la cadena seleccionada será determinada por Z_1 y $(1 - P_{\text{elección}})$ que será determinado por Z_2 . El efecto del VEGA es el mismo que el de usar una función

objetivo que combina linealmente los objetivos ponderados. Los pesos son determinados por la función de distribución de la población con respecto a los objetivos. El método VEGA es muy sensible al $P_{\text{elección}}$ que se utilice.

– Ordenación del óptimo de Pareto:

Se evalúa el valor de la cadena para cada objetivo. Si para una cadena dada no existen otras cadenas con mejores valores para los objetivos, esa cadena tendrá un orden asociado uno. Las cadenas con orden uno se remueven de la población y las que quedan se evalúan de igual forma que en el proceso anterior y se les da un orden de dos a las mejores cadenas. Este proceso continúa hasta que todas las cadenas son ordenadas. Las cadenas con menor orden eliminan a las de mayor orden.

– La combinación de estos dos últimos.

El proceso se da en dos pasos: en el primero se seleccionan N generaciones por el método VEGA. En el segundo paso N generaciones se ordenan por el método de ordenamiento del óptimo de Pareto. El método VEGA permite al usuario forzar la búsqueda hacia una porción particular de la curva de Pareto y el otro método permite encontrar el óptimo pasando de uno en uno dentro de la curva de Pareto.

4.8 Ventajas y desventajas de los Algoritmos Genéticos

Entre las ventajas de los algoritmos genéticos están:

- Permite funciones objetivo altamente no lineales.
- La búsqueda es guiada por información de valores de la función objetivo, que no necesita ser continua, diferenciable o unimodal. Esto es lo opuesto a los métodos de búsqueda basados en derivadas, como el método del gradiente.
- A diferencia de los otros métodos meta heurísticos, los AG buscan en una población y no en un único punto. Esto incrementa la probabilidad de que no quede atrapado en un óptimo local.
- Permite trabajar en paralelo para encontrar a la vez un grupo de soluciones, a diferencia de la mayoría de los métodos clásicos en los que se genera una solución a la vez.

- Permite buscar soluciones en espacios de decisión multimodales complejos.
- Se comporta eficientemente con no convexidades, que generalmente no son manejables por los métodos de optimización tradicionales.
- Usa la función objetivo en si misma y no usa información derivada.
- Usa reglas probabilísticas en vez de determinísticas como la mayoría de métodos tradicionales.

El método AG presenta también ciertas desventajas, pero la mayoría de ellas han sido evitadas con el uso de ciertas técnicas. Algunas son:

- Como tratar las restricciones. Esto se ha resuelto permitiendo que individuos no factibles pasen a la próxima generación, penalizando su bondad del ajuste en función de la violación a las restricciones como se explicó anteriormente. Una buena estrategia es permitir en la población inicial únicamente cadenas que cumplen las restricciones.

Para algunos problemas puede ser relativamente fácil alterar soluciones no factibles para convertirlas en factibles. Si ellas son transformadas mediante operadores reparadores pueden volverse útiles en la población. Los reparadores son efectivos en los casos en que la factibilidad sea difícil de conseguir. Sin embargo dependiendo de la forma en que tales operadores sean usados, pueden causar pérdida de diversidad en la población.

- Otro problema frecuente es la pérdida de diversidad de la muestra, lo que puede conducir a obtener prematuramente solo óptimos locales al no alcanzar a explorar todo el espacio factible. El apareamiento indiscriminado puede ser poco efectivo. Si se trata de optimizar una función multimodal, el cruzamiento de dos cromosomas que son muy cercanos a dos óptimos puede ser mucho peor que ellos.

Estimación de parámetros: Los parámetros necesarios suelen ser demasiados como tamaño de la muestra, $P_{\text{elección}}$, $P_{\text{mutación}}$ entre otros. Golberg (1989) indicó que el tamaño óptimo para la población de cadenas con código binario crece exponencialmente con la longitud de la cadena. Algunos autores opinan, basados empíricamente, que treinta es un tamaño de población adecuado (Schaffer, 1989).

Para profundizar en los fundamentos, características y tipos de algoritmos genéticos puede consultarse (Goldberg, 1989), (Fang, 1994) y en la parte biológica (Smith, 1979).

En este capítulo se han explicado los fundamentos de los algoritmos genéticos y las razones que los hacen la técnica más apropiada para la programación de itinerarios para el transporte público en Colombia.

El siguiente capítulo explica las características del algoritmo genético que se diseñó e implementó para solucionar el problema de la programación eficiente de rodamientos.

CAPITULO 5

ALGORITMO GENETICO PARA LA PROGRAMACION DE RODAMIENTOS O ITINERARIOS

Los algoritmos genéticos son ampliamente utilizados en tareas de optimización y específicamente en la programación de vehículos de Transporte Público Colectivo Urbano (Nasser, 2001), debido a que es un problema complejo que no siempre puede resolverse con técnicas de optimización convencionales. Es común encontrar en la literatura modelos de algoritmos genéticos y metaheurísticas utilizados para resolver problemas complejos de optimización, pero ninguno de ellos es aplicable al caso en estudio, dadas las condiciones particulares del sistema en Colombia. Se diseñó entonces un modelo de optimización para rodamientos de vehículos de Transporte Público Colectivo, aplicable a las ciudades colombianas, considerando sus criterios y restricciones específicas. Para dicho modelo se desarrolló un completo algoritmo genético híbrido que combina los mecanismos de los algoritmos genéticos con otras heurísticas, como se detallará a continuación.

Al igual que en la programación manual, el primer paso consiste en generar la tabla de despachos, para lo cual es necesario tener datos como: duración de recorridos, horario de servicio, tiempo de Terminal e intervalos de despacho. Se diseñará el rodamiento para la misma ruta piloto que se explicó en el Capítulo 3.

5.1 Esquema de Codificación

Cada cromosoma representa la programación completa para un día de trabajo. Cada uno de los genes corresponde a cada una de las horas de despacho de acuerdo a las

frecuencias asignadas para la ruta y el valor que se le asigna corresponde a la rutina que debe cubrir ese despacho.

No se utilizó codificación binaria, más bien, en cada uno de los genes se codificó directamente en decimal el número de la rutina que cubre ese despacho.

Para ilustrar la estructura del algoritmo, supóngase que se va a programar una ruta que se despacha desde las 5:30 a.m. hasta las 6:30 a.m. cada 30 minutos, desde las 6:30 a.m. hasta las 8:10 a.m. cada 20 minutos y desde las 8:10 a.m. hasta las 11:10 a.m. cada hora. Entonces la tabla de despachos quedaría como la Tabla 7.

Tabla 7. Codificación del algoritmo genético

Hora despacho	Gen No.
5:30 a.m.	1
6:00 a.m.	2
6:30 a.m.	3
6:50 a.m.	4
7:10 a.m.	5
7:30 a.m.	6
7:50 a.m.	7
8:10 a.m.	8
9:10 a.m.	9
10:10 a.m.	10
11:10 a.m.	11

Cada uno de los despachos constituye un gen, así que el cromosoma tendrá 11 genes y quedará conformado por los vehículos o rutinas asignadas a cada uno de los despachos. Si el cromosoma fuera el de la Tabla 8, indicaría que el vehículo o rutina 1 está asignado para el despacho de las 5:30 a.m., la rutina o vehículo 3 para el despacho de las 6:00 a.m., la rutina o vehículo 1 para el despacho de las 6:30 a.m. y así sucesivamente. El cromosoma tendrá un número de genes igual al número de despachos que se realizan durante un día. La codificación de los genes se hace en

decimal, asignándole el número correspondiente a la rutina que se programa para ese despacho.

Tabla 8. Ejemplo de un cromosoma

1	3	1	2	3	1	2	3	1	3	2
Gen1	Gen2	Gen3	Gen4	Gen5	Gen6	Gen7	Gen8	Gen9	Gen10	Gen11

Se habla de rutina porque lo que se está generando es una programación ideal que no va a cambiar de un día a otro. Más bien en un proceso posterior, esas rutinas se asignarán a diferentes vehículos de acuerdo al día de la semana.

El proceso de asignación de rutinas a cada uno de los genes o despachos, se hizo primero de forma aleatoria, pero no se obtuvieron resultados factibles debido a una restricción física que no se puede violar. Esta restricción es que no se puede despachar un vehículo antes de que haya concluido el despacho anterior. Entonces una generación aleatoria produce cromosomas no factibles. Por tal motivo se creó una estructura para almacenar los vehículos disponibles para cada uno de los despachos y antes de realizar cada asignación se consulta si existen rutinas o vehículos disponibles para ese despacho. Si no hay disponibles, se asignará un nuevo valor aleatoriamente. Las rutinas o vehículos disponibles son las que han hecho el recorrido completo y se pueden volver a programar para el siguiente despacho.

Por ejemplo: para el primer despacho no hay disponibles, entonces el sistema genera un número aleatorio de rutina o vehículo, en este caso el 1. Suponiendo que el tiempo de recorrido es 1 hora, la rutina o vehículo 1 estará disponible para volver a despacharse a las 6:30 a.m. que corresponde a la hora de despacho del tercer gen, entonces se inscribe en el archivo de disponibles del tercer gen, es decir, está disponible para el despacho a las 6:30 a.m. y como para esa hora no hay más vehículos disponibles, es la rutina o vehículo elegida para ese gen. Para el segundo despacho (6:00 a.m.) no hay disponibles, así que se asigna una rutina o vehículo al azar, en este caso la rutina o vehículo 3, esta rutina o vehículo estará disponible a las

7:00 a.m., por tanto, se agrega a las disponibles para el quinto gen (7:10 a.m.). El tercer despacho tiene como disponible la rutina o vehículo 1 y se asigna esa rutina, que queda disponible de nuevo para las 7:30 a.m., es decir para el sexto despacho. Para el cuarto despacho no hay disponibles, entonces se asigna aleatoriamente, en este caso es la rutina 2, que queda disponible para las 7:50 a.m., el séptimo despacho. Para el quinto despacho se asigna la 3 que es la única disponible, igual para el sexto la 1 y para el séptimo la 2. Para el despacho de las 9:10 a.m. hay tres rutinas disponibles, así que se escoge una aleatoriamente. Para el despacho de las 10:10 a.m. quedan dos disponibles y se escoge una de ellas. Finalmente para el último despacho está disponible la rutina o vehículo 2 y esa es la que se asigna. En la Tabla 9 se resumen los despachos asignados a cada uno de los genes, las rutinas disponibles y los tiempos de espera.

Tabla 9. Rutinas asignadas y disponibles para cada despacho

Hora despacho	Gen	Rutinas disponibles	Rutina asignada	Tiempo de espera
5:30 a.m.	1		1	
6:00 a.m.	2		3	
6:30 a.m.	3	1	1	
6:50 a.m.	4		2	
7:10 a.m.	5	3	3	10 min
7:30 a.m.	6	1	1	
7:50 a.m.	7	2	2	
8:10 a.m.	8	3	3	
9:10 a.m.	9	1,2,3	1	40 min
10:10 a.m.	10	2,3	3	1 hora
11:10 a.m.	11	2	2	2:20 min

Debido a la complejidad del problema no se puede utilizar un algoritmo genético tradicional, con codificación binaria y operadores de cruzamiento y mutación convencionales. Por tal motivo se diseñó e implementó un algoritmo genético híbrido que incorpora el conocimiento específico del problema no solo en la codificación, sino también en los operadores genéticos.

5.2 Población Inicial

Primero se ensayó la generación de cada uno de los cromosomas de la población aleatoriamente. Para tal fin se estableció un número fijo de rutinas y para cada uno de los despachos se escogió aleatoriamente una de esas rutinas. El grave problema de este mecanismo de generación es que no se obtuvieron soluciones factibles, en el sentido de que para poder despachar de nuevo un vehículo hay que esperar que haya transcurrido el tiempo de recorrido. Esta es una restricción física que no se puede violar, por lo cual es necesario mirar si el vehículo que se va despachar es un vehículo o rutina factible (en el sentido de haber concluido su despacho anterior y estar disponible para un nuevo despacho). En el ejemplo que se está considerando, si para el primer despacho se escogió el vehículo o rutina 1 y el tiempo de recorrido es 1 hora, no se puede volver a programar ese vehículo o rutina en el cromosoma hasta el gen 3.

Se hicieron varias pruebas utilizando la generación aleatoria de los individuos de la población sin ninguna restricción y sin utilizar mecanismos de reparación. Al generar 100.000 individuos, las infactibilidades producidas por despachar un vehículo antes de haber concluido su despacho anterior estuvieron entre 64 y 101 para todos los individuos, el mejor resultado fue 45 violaciones de 147 despachos. Para 10.000 individuos, las violaciones estuvieron entre 67 y 100, el mejor resultado fue 47. Y para 300.000 individuos, entre 63 y 99, siendo el menor número de violaciones 53. Estos resultados indican que se presenta un altísimo porcentaje de despachos no factibles y muestran la necesidad de utilizar un mecanismo de reparación o una heurística que garantice que las soluciones generadas son todas factibles.

Se podría pensar que a medida que el algoritmo va evolucionando se puede llegar a soluciones factibles pero después de 200 generaciones con 200 individuos cada una, apenas se llega a un mínimo de 49 violaciones por cromosoma y ninguna de las soluciones es factible por violar la restricción física de no volver a despachar un vehículo antes de haber concluido su despacho anterior.

Otra opción que también se ensayó, es predeterminar el número de vehículos o rutinas que se usarán en la programación. Cuando se empieza la programación, se colocan todas las rutinas como disponibles y cada vez que se va a realizar un despacho, se

escoge aleatoriamente entre esas rutinas disponibles. Para el primer gen todas las rutinas están disponibles, y se elige una aleatoriamente entre ellas, para el siguiente despacho estarán disponibles todas menos la que se acaba de programar y así sucesivamente. Este método tiene varios inconvenientes: uno de ellos es que se generarán muchas programaciones aparentemente diferentes que en realidad corresponden a la misma programación, pero cambiando el número de la rutina o vehículo. El otro inconveniente es cómo determinar cuál es el número de vehículos que se debe utilizar. Si se asignan muchos vehículos, se generarán muchísimas rutinas y luego será necesario unirlas para reducir el número de vehículos necesarios en la ruta. Se consideró más conveniente dejar el número de rutinas como uno de los criterios de la función de aptitud, en lugar de predeterminarlo como una restricción física del sistema.

Finalmente se decidió generar la población inicial de la siguiente forma: inicialmente no hay vehículos disponibles. Si no hay ningún vehículo disponible para despachar se incluye un nuevo vehículo a la lista. Cuando existan varios vehículos disponibles para un despacho, se seleccionará uno de ellos aleatoriamente. Una vez se asigna algún vehículo se actualiza la lista de los que quedan disponibles para la próxima asignación. El que fue asignado de último solo podrá aparecer en la lista nuevamente después de que transcurra el tiempo de recorrido. Este método es el que se utilizó en las mejoras a la programación manual que se explicaron en el capítulo 3. Esta codificación garantiza que la población inicial solo contendrá cromosomas factibles, en el sentido de que ningún vehículo aparecerá programado sin haber concluido su despacho anterior.

También se ensayó la opción de generar la población inicial escogiendo el vehículo que al ser despachado tuviera el menor tiempo de espera, es decir, el método final de la programación con heurística. Sin embargo, este mecanismo de generación tan fuerte hace converger rápidamente el algoritmo y no favorece la exploración del espacio del problema, por lo cual no se utilizó esta heurística en el algoritmo final.

5.3 Operadores Genéticos

Los operadores genéticos convencionales son el cruzamiento y la mutación. Pero cuando se utilizan los operadores con los individuos de la población, se presenta un grave inconveniente y es que en la mayoría de los hijos (por no decir en todos) se presentan infactibilidades, porque al unir una parte de un individuo factible con otra parte de otro individuo factible, pueden presentarse, especialmente cerca del punto de unión, rutinas que queden programadas antes de que hayan terminado su recorrido anterior completo. Esa infactibilidad física es intolerable en el sistema y es necesario resolverla de alguna forma. Para solucionar este inconveniente hay varias opciones: una es utilizar operadores de reparación, que conviertan las soluciones no factibles en soluciones factibles, y la otra es utilizar operadores genéticos especiales que solo generen soluciones factibles. En el algoritmo se utilizó la primera opción: operadores de reparación.

El operador de mutación que se emplea en el algoritmo, escoge uno de los vehículos o rutinas del cromosoma actual, lo elimina de la programación y lo reemplaza por otro escogido al azar. El vehículo asignado puede generar una solución no factible, si ese vehículo tiene otra programación en un tiempo menor al tiempo de duración del recorrido. Por lo tanto se diseñó un operador de reparación que se encarga de verificar si alguno de los vehículos o rutinas se está despachando antes de estar disponible para hacerlo, y cada vez que encuentra una infactibilidad de este tipo genera una lista de los vehículos o rutinas que podrían ser despachados a esa hora (disponibles) y escoge uno de ellos aleatoriamente para reemplazarlo por el gen no factible.

Para el operador de cruzamiento se utilizaron dos variantes: cruzamiento uniforme, que no da muy buenos resultados y el cruzamiento de un punto, que arrojó mejores resultados, por lo cual fue el que finalmente se implementó.

El cruzamiento uniforme que se ensayó funciona de la siguiente forma: se toman dos padres y se crea una máscara binaria (0 y 1) aleatoriamente, del tamaño del cromosoma. Si el valor de la máscara es cero, se toma el primer gen del primer padre, de lo contrario se toma del otro padre. Se hace lo mismo para todos los genes.

Después se requiere revisar el cromosoma hijo para ver si tiene infactibilidades, que se corrigen con el mismo mecanismo reparador utilizado en la mutación. Con este tipo de cruzamiento los resultados son muy deficientes, porque hay demasiadas infactibilidades.

Los ensayos con el operador de cruzamiento uniforme muestran que este mecanismo no es apropiado debido a la gran cantidad de violaciones de la restricción de no volver a programar un vehículo antes de que haya terminado su recorrido anterior. Al probar el algoritmo con este mecanismo, se pierden las características de los padres y las poblaciones de las siguientes generaciones se vuelven producto del mecanismo de reparación. No se observa un aumento en los valores de la función de aptitud del algoritmo. Por eso se reemplazó este operador por un cruzamiento de un punto.

El cruzamiento de 1 punto toma un gen aleatorio y en ese punto combina los genes de los dos padres. Una vez efectuado el cruzamiento se utiliza el mismo operador de reparación que se usó en la mutación. Si en alguno de los genes de los hijos queda un vehículo o rutina no factible, se reemplaza por otro escogido aleatoriamente de la lista de disponibles para ese gen.

El método para la selección de los padres asigna una probabilidad de reproducción a cada uno de los individuos de la población, dando mayor probabilidad a los individuos con mayor aptitud. Para tal fin se utilizó el método de la ruleta.

Inicialmente se hicieron pruebas sin elitismo, pero los valores de la función de valoración varían demasiado de una generación a otra. Entonces se vio la necesidad de incorporar elitismo, pasando a la siguiente generación los mejores dos individuos de la generación anterior.

5.4 Evaluación de la población

A cada cromosoma se le debe medir su aptitud. La función de aptitud (fitness) depende de varios criterios, algunos de los cuales, si se trabajara con programación matemática, serían restricciones pero como aquí se pueden violar, lo que se hace es

considerarlos como criterios que se penalizan de alguna manera cuando no se cumplen estrictamente.

Esos criterios estarán midiendo diferentes cosas, pueden ser conflictivos entre sí y no necesariamente están expresados en las mismas unidades de medida. Algunos son criterios que se deben maximizar y otros se deben minimizar. Es necesario entonces llevarlos a las mismas unidades y para eso se usa una función de pertenencia difusa para cada uno de los criterios que debe cumplir las siguientes condiciones (considerando criterios que se desean maximizar; si se desea minimizar se hace lo contrario):

- Debe asignar a la mayor valoración del criterio el valor de 1 y a la menor valoración del criterio el valor 0.
- Valoraciones que estén por debajo de un límite de admisibilidad $L1_i$ fijado por los decisores, deben tomar el valor de 0. Este límite indicará que las alternativas con valoraciones menores a ese valor son inadmisibles
- Valoraciones que superen un valor $L2_i$, fijado por los decisores, deben tomar el valor de 1. Este límite indicará que las alternativas con valoraciones mejores a $L2_i$ cumplen las metas (considerando únicamente el criterio en cuestión).
- A las valoraciones intermedias entre $L1_i$ y $L2_i$ asignar valores entre 0 y 1. Según la importancia del criterio, la función de reescalamiento puede ser cóncava, convexa o lineal (generalmente se usa lineal). Este reescalamiento indicará que tan importante es o no la diferencia entre valoraciones específicas del criterio, ya que la pendiente puede cambiar a lo largo del rango $L1_i - L2_i$

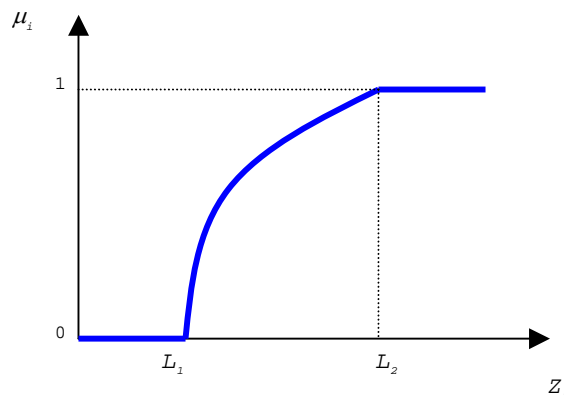


Figura 7. Función de pertenencia de un criterio

Luego de establecer los criterios y los rangos para la valoración, se procede a obtener la valoración para cada uno de los criterios que estarán ahora en las mismas unidades y representados por un valor entre 0 y 1.

Para obtener la función de adaptación (*fitness*) de cada cromosoma se pueden utilizar varios métodos:

1. Tomar como valor para la función de aptitud, al peor de los valores obtenidos en los diferentes criterios. Por ejemplo, si un cromosoma tiene $\mu_1=0.1$, $\mu_2=0.2$, $\mu_3=0.51$ (Para los diferentes criterios), el valor de aptitud del cromosoma será 0.1, que corresponde al peor valor de los tres criterios. Esta valoración es muy radical porque solo se fija en el peor defecto de la cadena.
2. Utilizar la siguiente función:

$$(5.1) \quad I_j = \left[\sum_{i=1}^p [w_i (1 - \mu_{ij})]^2 \right]^{1/2}$$

Donde j es el cromosoma evaluado, p es el número de criterios y w_i los pesos que determinan la importancia de cada criterio y que deben sumar en total 1. Entre menor sea el valor de la función I_j mayor será la aptitud del cromosoma en cuestión, por lo tanto se debe usar el inverso de este valor.

El segundo método tiene varias características deseables: la distancia entre 1 y μ_{ij} , define el denominado “arrepentimiento”, es decir distancia al ideal normalizado 1. El exponente 2 castiga fuertemente a las alternativas con valores extremadamente desventajosos en algunos de los criterios. Bajo esta formulación puede resultar mejor valorada una alternativa con valores medios en todos los criterios que una alternativa con excelentes valores en unos criterios y pésimos valores en otros.

En el algoritmo genético se utilizó el segundo método, por las ventajas que ofrece en la valoración multi-objetivo.

Los criterios más importantes para calificar una programación de itinerarios son el tiempo de espera en Terminal, el número de vueltas que da cada una de las rutinas o

vehículos, los descansos o almuerzos y la hora en la que se programan y el número de rutinas o vehículos necesarios para la programación. Otros criterios, tales como la duración de la jornada laboral, están implícitos en los anteriores. Por lo tanto los criterios que forman parte de la función de aptitud en la Ecuación (5.1) son:

Criterio 1: Este criterio busca que la programación tenga el mismo número de vueltas o recorridos para cada una de las rutinas. El objetivo es mantener la equidad en la programación, que no le toque mucho trabajo a unos vehículos y muy poco a otros. Esta relacionado con el promedio de vueltas, pero penaliza más fuertemente a los que se alejan demasiado del promedio.

Minimizar:

$$(5.2) \quad C_1 = \sum_i \sqrt{(Vueltas(i) - Promedio_Vueltas)^2}$$

Donde:

i es cada una de las rutinas

Promedio_vueltas es la suma de las vueltas que da cada una de las rutinas del cromosoma actual, dividida entre el número de rutinas.

Como se requiere minimizar este criterio y que su valoración quede entre 0 y 1, se calcula el valor máximo y valor mínimo para el criterio dentro de la población actual y se aplica la siguiente fórmula:

$$(5.3) \quad C'_1 = \frac{C_1 \max - C_1}{C_1 \max - C_1 \min}$$

Donde:

C'_1 es el criterio normalizado entre 0 y 1.

C_1 max es el valor máximo del criterio en la población actual

C_1 min es el valor mínimo del criterio en la población actual

Criterio 2: Este criterio trata de minimizar el número de vehículos o rutinas utilizados en la programación. En el denominador, al número de rutinas se le resta el menor número de rutinas de la población actual, para no obtener números muy pequeños en

la valoración de este criterio. El valor ideal es 1 cuando se utiliza el mínimo número de vehículos.

Maximizar:

$$(5.4) \quad C_2 = \frac{1}{(1 + \text{Número_Rutinas} - \text{Mín_Num_Rutinas})}$$

Donde:

Número_Rutinas es el número de rutinas o vehículos del cromosoma actual

Min_Num_Rutinas es el mínimo número de rutinas o vehículos de la población actual

Criterio 3: Es uno de los criterios más importantes porque busca minimizar los tiempos de espera de cada una de las rutinas. El tiempo que se da para almorzar o el descanso extra, no se tienen en cuenta como tiempo de espera. Este tiempo de espera, es el tiempo que el vehículo está en el Terminal después de haber concluido su último recorrido y antes de ser despachado de nuevo.

Minimizar:

$$(5.5) \quad C_3 = \sum_i \text{esperas}(i)$$

Donde:

i es cada una de las rutinas

esperas(i) es la suma de las esperas de la rutina *i* en cada una de las vueltas.

Para que la valoración de este criterio esté entre 0 y 1 se aplica de nuevo la Ecuación (5.3) para el criterio C_3 utilizando los valores máximo y mínimo del criterio en la población actual.

Criterio 4: Este criterio se relaciona con los almuerzos. Premia a las rutinas que tienen tiempo para almorzar y las valora teniendo en cuenta la hora a la cual se

programó el almuerzo. Es un hecho que hay ciertas horas del día que los conductores prefieren para almorzar.

Hay un parámetro que el usuario establece y determina el tiempo mínimo que se considera aceptable para almorzar. En las pruebas se utilizó un valor de 30 minutos. Cuando alguna rutina tiene un tiempo de espera superior a ese parámetro, entonces esa primera espera se convierte en el almuerzo.

La valoración se hace de la siguiente forma:

Si el vehículo no tiene asignado tiempo para almuerzo, entonces: Alm=0

Si tiene tiempo asignado para almorzar:

Antes de las 11:00 entonces: Alm = 0.2

Entre las 11:00 y las 12:00, entonces: Alm=0.2 + 0.8*(hora-11)

Entre las 12 y las 15:00, entonces: Alm = 1

Entre las 15 y las 16:30, entonces: Alm = 1 – 0.9*(hora - 15)/1.5

Después de las 16:30, entonces: Alm = 0.1

Esta valoración obedece a lo que expresan los conductores con respecto a la hora que consideran más conveniente para almorzar. Ellos dicen que el almuerzo ideal debería ser entre las 11 de la mañana y las 3 de la tarde por lo que se asigna el valor máximo en ese horario. Pero si no es posible asignarles almuerzo en esa hora la valoración de este criterio será inferior. El criterio queda entonces como:

Maximizar:

$$(5.6) \quad C_4 = \frac{\sum_i Alm(i)}{Num_Rutinas}$$

Donde:

i es la rutina

Alm es la valoración del almuerzo

$Num_Rutinas$ es el número de rutinas

Pesos: En la ecuación (5.1) cada uno de los criterios es multiplicado un peso W_i que indica que tan importante es ese criterio en la valoración final. El valor de este

parámetro se puede escoger en el programa, lo importante es que la suma de pesos sume 1. Para la mayoría de las pruebas se escogió una valoración de 0.25 para cada uno de los criterios, indicando que cada uno de ellos tiene la misma importancia. El cambio en esos pesos afecta considerablemente los resultados (Ver Capítulo 6).

Las empresas de transporte público pueden tener una valoración de pesos diferentes. Para algunos lo más importante es reducir el tiempo de espera, para otros puede ser más importante tener el menor número de vehículos trabajando y para otra empresa puede ser más importante que los conductores tengan tiempo para almorzar. Por esa razón se dejan esos parámetros configurables antes de la ejecución del algoritmo.

5.5. Algoritmo genético para la programación de rodamientos

El código fuente del programa que implementa el Algoritmo Genético aparece en el Anexo 9. El programa se desarrollo en Borland Delphi 7.0. El instalador del programa se anexa en el CD ROM adjunto.

5.6 Asignación de la programación semanal

Una vez que se obtiene la programación óptima del rodamiento para un día, es necesario asignar los vehículos de la empresa que cubrirán cada una de las rutinas a lo largo de la semana. Los fines de semana se programan de la misma manera, pero las frecuencias de despacho y tiempos de recorrido son diferentes, lo cual genera programaciones que requieren menos vehículos. Las asignaciones de fin de semana se manejan aparte de las semanales, ya que no es lo mismo cubrir una ruta un lunes, que un domingo.

La asignación de buses a las diferentes rutinas durante la semana debe hacerse considerando que debe buscar el equilibrio, es decir, que a los buses se les asigne en la semana un número similar de rutinas. Además, como no todas las rutinas son igualmente convenientes para los conductores y propietarios de los buses, la programación debe tratar de asignar considerando equidad en el reparto. Como no todos los buses se requieren, habrá dificultad en asignarle rutinas a todos. Solamente

considerando un número adecuado de buses activos, es posible hacer asignaciones económicamente convenientes para todos. La asignación se realizará mediante la optimización del problema que se detallará a continuación:

Consideremos a x_{ijk} como la variable de decisión binaria que identifica a la asignación o no del bus i a la rutina j en el día k . Los subíndices serán entonces:

$i = 1, \dots, m$, donde m es el número de buses activos de la empresa

$j = 1, \dots, n$, donde n es el número de rutinas diarias encontradas mediante el algoritmo híbrido AG

$k = 1, \dots, 5$, y se refiere al día de la semana.

La función objetivo busca el equilibrio de despachos entre todos los buses, así:

Minimizar (Máxima cantidad de despachos asignados en toda la semana a alguno de los buses - Mínima cantidad de despachos asignados en toda la semana a alguno de los buses)

Y deben cumplirse las siguientes restricciones:

- Para cada rutina j y cada día k puede ser asignado solo un bus. Esta restricción puede expresarse como:

$$(5.7) \quad \sum_i x_{ijk} = 1, \text{ para todo } (j,k)$$

- Para cada bus i y cada día k puede asignarse máximo una rutina. Esta restricción puede expresarse como:

$$(5.8) \quad \sum_j x_{ijk} \leq 1, \text{ para todo } (i,k)$$

- A cada bus se le asignarán rutinas máximo durante 4 días de la semana:

$$(5.9) \quad \sum_j \sum_k x_{ijk} \leq 4, \text{ para todo } i$$

El objetivo busca equilibrar el número de rutinas asignadas a los buses. Esta función no es lineal, pero fácilmente puede linealizarse de la siguiente manera:

$$(5.10) \quad \text{Minimizar } (\beta - \lambda)$$

Donde β es el máximo número de rutinas asignadas durante la semana a alguno de los buses y λ es el mínimo número de rutinas asignadas durante la semana a alguno de los buses. E incorporando en el conjunto de restricciones original, las siguientes restricciones:

Para cada bus i :

$$(5.11) \quad \sum_j \sum_k x_{ijk} \leq \beta$$

$$(5.12) \quad \sum_j \sum_k x_{ijk} \geq \lambda$$

donde el termino de la izquierda de cada restricción indica la cantidad de rutinas que se le asigna a cada bus en la semana.

Mientras el número de buses activos sea excesivo, λ será 0. El máximo número de buses que deben estar activos se obtiene cuando m permita obtener $\lambda > 0$.

En la formulación anterior no se tuvo en cuenta la conveniencia de las diferentes rutinas para los conductores o empresarios. Esta conveniencia depende de lo adecuado o no del tiempo asignado para almorzar, del número de vueltas, de los descansos entre vueltas y de la duración de la jornada completa.

Si $p_{r_{jk}}$ representa un índice de preferencia entre 0 y 1 de la rutina j en el día k , una asignación más justa se obtendría de incorporar en las restricciones adicionales ese factor, así:

$$(5.13) \quad \sum_j \sum_k pr_{jk} \times x_{ijk} \leq \beta$$

$$(5.14) \quad \sum_j \sum_k pr_{jk} \times x_{ijk} \geq \lambda$$

De esta forma, el factor que se desea equilibrar no es la suma de rutinas asignadas durante la semana sino la calificación total de las rutinas asignadas durante toda la semana.

El problema completo queda entonces de la siguiente manera:

$$(5.15) \quad \text{Minimizar } (\beta - \lambda)$$

Sujeto a:

$$(5.16) \quad \sum_i x_{ijk} = 1, \text{ para todo } (j,k)$$

$$(5.17) \quad \sum_j x_{ijk} \leq 1, \text{ para todo } (i,k)$$

$$(5.18) \quad \sum_j \sum_k x_{ijk} \leq 4, \text{ para todo } i$$

$$(5.19) \quad \sum_j \sum_k pr_{jk} \times x_{ijk} - \beta \leq 0, \text{ para todo } i$$

$$(5.20) \quad \sum_j \sum_k pr_{jk} \times x_{ijk} - \lambda \geq 0, \text{ para todo } i$$

x_{ijk} , variable binaria

El índice pr_{jk} es el resultado de normalizar la calificación de la rutina j del día k según los factores mencionados anteriormente.

La aplicación de este algoritmo genético a la programación de la ruta piloto, así como la comparación con diferentes programaciones clásicas manuales, se presenta en el siguiente capítulo. Además también se presentan los resultados de la asignación semanal.

CAPITULO 6

CASOS DE APLICACIÓN Y RESULTADOS

En este capítulo se presentan los resultados obtenidos al comparar dos programaciones elaboradas con el método manual y la programación obtenida al aplicar los algoritmos genéticos. Se tomó como caso de aplicación la programación de la ruta Estambul-Universidad-Estambul cuyos parámetros se explicaron en la sección 3.4. Al final del capítulo aparecen los resultados de la asignación semanal de rutinas, este es un proceso posterior a la programación que consiste en asignar las diferentes rutinas a cada uno de los vehículos para los diferentes días de la semana.

6.1 Programación Manual de la ruta piloto.

Con los datos de la Sección 3.4, se elabora la tabla de despachos (Ver Tabla 5). Se generan 147 filas correspondientes a cada uno de los despachos diarios de la ruta. Con el método manual (Molinero y Sánchez, 1998) se van asignando consecutivamente los vehículos asignados a la ruta. Para este caso se requieren como mínimo 19 vehículos para cubrir los diferentes despachos.

En el Anexo 1 se observa la programación completa utilizando 19 vehículos. El Anexo 2 presenta esa programación organizada por rutinas. Igualmente, en la Tabla 10 se presenta un resumen de esa programación. La primera columna corresponde a la rutina, la siguiente es el numero de vueltas o recorridos completos, la tercera columna es el acumulado de las esperas adicionales a los 5 minutos de Tiempo de Terminal por recorrido. La columna jornada laboral corresponde al tiempo transcurrido desde el primer despacho hasta que termina el último despacho. La siguiente columna es la menor de las esperas de todos los recorridos, la siguiente la espera más alta, luego

aparece el promedio de esperas y finalmente aparece la columna tiempo/vuelta, que se calcula dividiendo la jornada laboral entre el número de vueltas o recorridos. En las últimas dos filas aparece el promedio y la desviación estándar de cada una de las columnas. La desviación estándar es una medida de qué tan dispersos están los datos del promedio y es un indicador importante que muestra que tan similares son los resultados para cada una de las rutinas. Así que entre menor sea la desviación estándar, más equitativas serán las rutinas de la programación. A excepción del número de vueltas, entre menor valor tenga cada una de las columnas es mejor.

Tabla 10. Resultados de la programación manual con Tiempo de Terminal.

RUTINA	NUMERO VUELTAS	ESPERA TOTAL	JORNADA LABORAL	ESPERA MIN	ESPERA MAX	ESPERA PROMEDIO	TIEMPO/VUELTA
1	8	02:24	14:54	00:06	00:32	00:18	01:51
2	8	02:23	14:53	00:08	00:34	00:17	01:51
3	8	02:22	14:52	00:08	00:36	00:17	01:51
4	8	02:23	14:51	00:05	00:40	00:17	01:51
5	8	02:20	14:52	00:02	00:40	00:17	01:51
6	8	02:24	14:54	00:04	00:40	00:18	01:51
7	8	02:26	14:56	00:06	00:40	00:18	01:52
8	8	02:28	14:58	00:08	00:40	00:18	01:52
9	8	02:35	15:01	00:10	00:40	00:19	01:52
10	8	02:40	15:06	00:12	00:40	00:20	01:53
11	8	02:45	15:11	00:10	00:40	00:20	01:53
12	8	02:50	15:16	00:08	00:40	00:21	01:54
13	8	03:05	15:31	00:06	00:55	00:23	01:56
14	8	03:18	15:46	00:04	01:10	00:24	01:58
15	7	02:06	13:05	00:02	00:40	00:18	01:52
16	7	02:04	13:05	00:00	00:38	00:17	01:52
17	7	02:04	13:05	00:00	00:36	00:17	01:52
18	7	02:06	13:05	00:02	00:34	00:18	01:52
19	7	02:08	13:07	00:04	00:32	00:18	01:52
PROM	7,73	02:27	14:33	00:05	00:40	00:19	01:52
DES.EST.	0,4403474	00:19	00:54	00:03	00:08	00:01	00:01

La primera columna del Anexo 1 llamada “Vehículos Esperando” (Ver Anexo 1) dice cuántos vehículos se encuentran esperando en Terminal después de cada uno de los despachos. Uno de los problemas de esta programación es que, frecuentemente, varios vehículos se encuentran esperando para ser despachados, por ejemplo entre las 9:11 a.m. y las 11:05 a.m. permanecen esperando cinco vehículos en Terminal, igual ocurre entre las 3:31 p.m. y las 5:05 p.m. Estas esperas son un problema debido

a la ausencia de un Terminal de despacho apropiado, así que los vehículos que están esperando se quedan parqueados en la vía pública, generando perjuicio a la comunidad del sector donde se encuentra el Terminal de despacho. Para la programación completa de esta ruta se efectúan 147 despachos, de los cuales 34 despachos presentan cinco vehículos esperando en Terminal. Eso quiere decir que para el 23.12% de los despachos quedan cinco vehículos esperando en Terminal.

Hay que tener en cuenta que al tiempo de espera hay que agregarle los cinco (5) minutos de Tiempo de Terminal. Eso quiere decir que cuando el tiempo de espera es cero (0) minutos, en realidad el conductor tiene cinco (5) minutos para tomar un breve descanso. Este es un parámetro de diseño que no se debe pasar por alto, porque permite cierta flexibilidad en la programación en caso de que alguno de los vehículos sufra un pequeño retraso.

Si se elimina el tiempo de Terminal (los cinco minutos que se agregan al tiempo de recorrido), se puede efectuar esta programación con un vehículo menos. Sin embargo, al hacerlo se están violando los parámetros de diseño sugeridos en la literatura (Molinero y Sánchez, 1998) (Pine et. al., 1998). El Anexo 3 presenta la programación eliminando este parámetro y en el Anexo 4 se puede observar esta programación organizada por rutinas.

Esta programación utiliza un vehículo menos, pero ya no existe un margen de tiempo que permita manejar algunos retrasos, sin que se altere la programación. Esto quiere decir que cuando el tiempo de espera es cero (0) minutos, el vehículo llega y tiene que ser despachado inmediatamente, sin la posibilidad de tomar un breve descanso. Si ese vehículo se retrasa, aunque sea un minuto, no es posible que cumpla con el siguiente despacho. Esa situación se presenta en los despachos de las 7:40 a.m., 6:30 p.m., 6:35 p.m. y 6:40 p.m. (Ver Anexo 3) haciendo demasiado estricto el cumplimiento de la programación a horas pico, en las que se pueden presentar retrasos debidos a la congestión debida a un mayor tráfico vehicular. Un resumen de los resultados de esta programación se presenta en la Tabla 11.

Como ya se ha mencionado, la ventaja de esta programación es que se puede utilizar un vehículo menos, lo cual aumentaría el promedio de pasajeros movilizados por

vehículo, pero fuera de que se están violando los parámetros de diseño, tiene el inconveniente de requerir el estricto cumplimiento de los tiempos de recorrido, ya que cualquier retraso en un vehículo puede hacer que se incumpla la programación, a menos que se tengan suficientes vehículos de reserva. Eso quiere decir que el vehículo que se gana en la programación, se pierde por la necesidad de tener varios vehículos de reserva para cubrir los posibles retrasos en los despachos críticos. Esta programación también presenta el problema de que en varios períodos de tiempo se encuentran esperando en el Terminal de despacho 5 vehículos (entre las 9:11 a.m. y las 11:00 a.m., entre las 3:31 p.m. y las 5:00 p.m., y a las 8:00 p.m.). En total se presenta esta situación en 32 de los 147 despachos diarios (Ver Anexo 3), lo que equivale a decir que se presenta este inconveniente en el 21.76% de los despachos. También se puede observar que la jornada laboral de algunos vehículos supera las 16 horas, lo cual puede ser excesivo para un día de trabajo de un conductor.

Tabla 11. Resultados de la programación manual sin Tiempo de Terminal.

RUTINA	NUMERO VUELTAS	ESPERA TOTAL	JORNADA LABORAL	ESPERA MIN	ESPERA MAX	ESPERA PROMEDIO	TIEMPO/ VUELTA
1	9	03:11	16:16	00:12	00:40	00:21	01:48
2	9	03:12	16:28	00:02	00:55	00:21	01:49
3	9	03:22	16:40	00:00	00:36	00:22	01:51
4	8	02:09	14:01	00:00	00:34	00:16	01:45
5	8	02:06	14:00	00:00	00:38	00:15	01:45
6	8	02:08	14:00	00:02	00:38	00:16	01:45
7	8	02:10	14:02	00:04	00:38	00:16	01:45
8	8	02:12	14:04	00:06	00:38	00:16	01:45
9	8	02:16	14:06	00:08	00:38	00:17	01:45
10	8	02:18	14:08	00:10	00:38	00:17	01:46
11	8	02:20	14:10	00:11	00:38	00:17	01:46
12	8	02:22	14:12	00:11	00:38	00:17	01:46
13	8	02:26	14:14	00:11	00:38	00:18	01:46
14	8	02:28	14:16	00:11	00:38	00:18	01:47
15	8	02:30	12:30	00:11	00:38	00:18	01:33
16	8	02:33	12:30	00:10	00:38	00:19	01:33
17	8	02:38	12:30	00:08	00:38	00:19	01:33
18	8	02:41	12:30	00:06	00:38	00:20	01:33
PROM	8,17	02:30	14:08	00:06	00:38	00:18	01:43
DES.EST.	0,372678	00:23	01:15	00:04	00:04	00:02	00:05

Otro problema adicional es que las rutinas, fuera de tener largas jornadas de trabajo, carecen del tiempo apropiado para tomar el almuerzo. Por ejemplo, en la Tabla 12 se

muestra la programación de la rutina 3, extraída del Anexo 4. Puede notarse que hay una espera de 32 minutos a las 9:00 a.m., demasiado temprano para tomar el almuerzo y una espera de 36 minutos a las 4:24 p.m. demasiado tarde para almorzar. Mientras que las esperas que se tienen en las horas apropiadas para almorzar son muy pequeñas. Este fenómeno es igual para todas las rutinas de esta programación, como puede constatarse en el Anexo 4. Esto quiere decir que esta programación carece de la programación apropiada de almuerzos, lo cual hace que muchos conductores abandonen su programación para tomar su almuerzo. Se supone que para almorzar se requiere un tiempo mínimo de 30 minutos.

Tabla 12. Programación de una rutina sin Tiempo de Terminal

Hora Despacho	Hora Disponible	Rutina	Espera (min)
05:56	07:22	3	00:08
07:30	09:00	3	<u>00:32</u>
09:32	11:00	3	00:25
11:25	12:55	3	00:15
13:10	14:40	3	00:16
14:56	16:24	3	<u>00:36</u>
17:00	18:30	3	00:00
18:30	20:00	3	01:10
21:10	22:36	3	
	Total	9	03:22

Las dos programaciones anteriores se han elaborado con el método manual clásico (Molinero y Sánchez, 1998) (Pine et. al., 1998), pero la realidad es que las empresas utilizan muchos más vehículos de los programados. Este fenómeno se debe a la gran sobreoferta que tienen las empresas. En un estudio real realizado en esta ruta se observó que actualmente se programan 22 vehículos para cubrirla (y eso que el 10% del parque automotor no se asigna por la restricción del “*pico y placa*”). Para entender más fácilmente la situación real, supongamos que una empresa tiene asignadas 4 rutas y que tiene 100 vehículos. Si las condiciones de operación de las 4 rutas son similares, se asignan 25 vehículos para cada ruta, independientemente de que la ruta se pueda cubrir de manera óptima con menos vehículos.

Pero más grave que asignar muchos más vehículos de los necesarios, es el hecho de que esos vehículos estarán en servicio durante todo el día, generando tiempos de espera excesivos y produciendo aglomeración en los terminales de despacho. Para visualizar mejor el problema se incluye el rodamiento para esta ruta elaborado manualmente con 22 vehículos que trabajan durante todo el día. En el Anexo 5 se presenta el rodamiento o itinerario completo para esta ruta piloto y en el Anexo 6 se organiza este rodamiento por rutinas o vehículos.

Lo primero que se observa al mirar el Anexo 6 es que las esperas en Terminal se vuelven muy grandes, inclusive se pueden observar varias esperas mayores a 1 hora. A medida que se utilizan más vehículos de los necesarios, las esperas aumentan y el número de vueltas por vehículo se reduce. Ambos factores son negativos y muestran una total ineficiencia en la programación.

El número de vehículos que permanece esperando en Terminal es muy elevado y esa es una de las razones por las cuales no se cumple la programación, ya que el despachador empieza a despachar a frecuencias diferentes a las programadas con el objeto de descongestionar el Terminal de despacho. En el estudio que se hizo sobre esta ruta se evidencia el total incumplimiento de las frecuencias de despacho, lo cual afecta la calidad del servicio.

Las largas jornadas sin un descanso programado lo suficientemente grande para el almuerzo y a una hora apropiada, hacen que los conductores abandonen los turnos para irse a almorzar, haciendo que a pesar de la sobre oferta de servicio, se presente un elevado incumplimiento de los despachos programados.

Otro problema que se evidencia son las largas jornadas de trabajo, por ejemplo, la rutina 11 trabaja desde las 5:40 a.m. hasta las 8:27 p.m., eso equivale a una jornada de trabajo de 14 horas con 47 minutos y solamente efectúa 7 vueltas. El tiempo de espera acumulado durante esa jornada es de 3 horas y 57 minutos. Los resultados son muy similares para el resto de rutinas.

Hasta el momento se observa que en la programación intervienen diferentes criterios que si se optimizan individualmente pueden generar rutinas muy buenas en el criterio que se está optimizando, pero deficientes en los demás. Por ejemplo, si no se despachan consecutivamente todos los vehículos, sino más bien se van introduciendo los vehículos en el momento que no haya otros disponibles, se logra una importante disminución en los tiempos de espera de cada una de las rutinas pero se generan diferencias muy grandes entre ellas. Por ejemplo, pueden quedar rutinas de 10 vueltas mientras que otras pueden tener apenas 2 vueltas.

Otro efecto negativo es que las rutinas buenas tienen tan poco tiempo de espera que los conductores de los vehículos no alcanzarían a almorzar y sería necesario reprogramarlas para incluir un tiempo suficiente de almuerzo.

Una cosa es importante aclarar: si se desea optimizar la programación no todos los vehículos deben estar en servicio durante toda el día. Los propietarios piensan que si el vehículo no está rodando van a perder dinero, pero en realidad si dejan de trabajar a ciertas horas pueden disminuir sus costos de operación, la contaminación, la congestión vehicular y aumentar los ingresos de los que si estén trabajando a esas horas.

6.2 Resultados de la programación con el Algoritmo Genético.

Para probar el algoritmo se requiere determinar la probabilidad de mutación, la probabilidad de cruzamiento, el tamaño de la población inicial, el número máximo de generaciones y los pesos de cada uno de los criterios. El programa permite al usuario cambiar cualquiera de los parámetros. A continuación se muestran los resultados que se obtienen con diferentes valores para algunos de los parámetros. Los pesos de los cuatro criterios se establecieron en 25% para darle a cada uno la misma ponderación. (Ver Tabla 13).

Después de realizar múltiples pruebas, se decidió utilizar los siguientes valores para los parámetros: cada uno de los cuatro criterios valorado con 25% (para que tengan el mismo peso), la tasa de mutación de 0.1%, la tasa de cruzamiento de 70%, una población inicial de 30 individuos y 30 generaciones. Los resultados obtenidos por el

algoritmo genético con estos parámetros permitieron obtener la programación que aparece completa en el Anexo 7 y organizada por rutinas en el Anexo 8. La Tabla 14 muestra un resumen de los resultados de la programación.

Tabla 13. Resultados del *Fitness* con diferentes valores para los parámetros

Mutación	Cruzamiento	Población	Generaciones	Mejor Fitness
0,001	0,7	100	50	13,11
0,001	0,7	20	20	11,35
0,05	0,7	10	20	11,36
0,05	0,3	20	25	13,28
0,01	0,5	10	25	9,6

Tabla 14. Resultados de la programación con el Algoritmo Genético.

RUTINA	NUMERO VUELTAS	ESPERA TOTAL	JORNADA LABORAL	ESPERA MIN	ESPERA MAX	ESPERA PROMEDIO	TIEMPO/ VUELTA
1	8	00:18	12:50	00:00	00:07	00:02	01:36
2	9	00:38	14:37	00:00	00:16	00:04	01:37
3	8	00:51	13:21	00:00	00:22	00:06	01:40
4	8	00:53	13:23	00:00	00:25	00:06	01:40
5	7	00:51	11:50	00:00	00:25	00:07	01:41
6	9	00:42	14:45	00:00	00:19	00:04	01:38
7	8	00:36	13:06	00:00	00:13	00:04	01:38
8	8	00:40	13:08	00:00	00:13	00:05	01:38
9	7	01:40	12:35	00:00	00:27	00:14	01:47
10	8	00:19	12:49	00:00	00:05	00:02	01:36
11	8	00:30	13:00	00:00	00:12	00:03	01:37
12	8	00:28	12:58	00:00	00:12	00:03	01:37
13	7	00:36	11:31	00:00	00:13	00:05	01:38
14	7	00:34	11:33	00:00	00:20	00:04	01:39
15	8	00:33	13:01	00:00	00:12	00:04	01:37
16	8	01:14	13:46	00:00	00:26	00:09	01:43
17	7	00:34	11:31	00:00	00:19	00:04	01:38
18	7	00:20	11:19	00:00	00:14	00:02	01:37
19	7	00:49	11:42	00:00	00:20	00:07	01:40
PROM	7,74	00:41	12:46	00:00	00:16	00:05	01:39
DES.EST	0,635949	00:19	00:58	00:00	00:06	00:02	00:02

Esta programación trata de asignar a cada una de las rutinas un tiempo programado para que el conductor pueda almorzar. Además busca que ese tiempo de almuerzo se presente a una hora conveniente. Para algunas de las rutinas se tienen dos tiempos

programados de descanso (uno para el almuerzo y otro adicional), ya que se reportan como rutinas deseables, las que tienen dos o tres franjas de trabajo (Pine et. al., 1998). Estos descansos programados no se incluyen como esperas en Terminal, ni dentro de la jornada laboral, porque se supone que el conductor puede retirarse del Terminal de despacho y regresar cuando finalice su tiempo de almuerzo o descanso. En la Tabla 15 se observan esos descansos y la hora a la que se presentan para cada una de las rutinas.

Tabla 15. Programación de almuerzos y descansos.

RUTINA	TIEMPO ALMUERZO	HORA ALMUERZO	TIEMPO DESCANSO	HORA DESCANSO
1	00:30	13:15	01:10	08:50
2	00:36	12:29	01:20	19:30
3	00:48	09:05		
4	01:21	14:45		
5	02:20	14:50		
6	00:31	11:19		
7	01:01	09:34		
8	01:08	09:41		
9	01:57	15:33		
10	01:32	09:48	01:45	19:25
11	00:54	13:20		
12	01:12	08:20		
13	02:30	15:15	01:10	19:20
14	01:32	10:23		
15	00:48	12:01	00:34	14:22
16	01:00	10:30		
17	02:35	15:30		
18	02:25	09:00		
19	02:19	16:01		

En la Tabla 15 se resaltan con negrita y cursiva los horarios apropiados para almorzar, eso quiere decir que 10 de las 18 rutinas pueden tener un tiempo de almuerzo de por lo menos 30 minutos a una hora razonable. Las programaciones elaboradas con el método clásico no tienen tiempo suficiente para almorzar en horarios apropiados. Además en vista de que una jornada laboral larga puede ser muy dura, en esta programación las esperas entre despachos son mínimas, pero se tiene un descanso de más de 30 minutos para almorzar y en algunas rutinas se tienen dos descansos (las rutinas que tienen valores en las dos últimas columnas de la Tabla 15). Por ejemplo la rutina 2 empieza a trabajar a las 5:48 a.m. hasta las 12:29 p.m., cuando sale de programación y tiene 36 minutos para el almuerzo. A la 1:05 p.m. inicia de

nuevo su trabajo hasta las 7:30 p.m. donde tiene un descanso de 1 hora y 20 minutos antes de su último recorrido (Ver Anexo 8).

Cuando se usa análisis multicriterio, como el empleado en el diseño del Algoritmo Genético, es imposible comparar la respuesta con la dada por otro método, sea de optimización o no, porque precisamente ahí radica parte del aporte: se consideran criterios que bajo otras ópticas también son importantes, pero que los métodos clásicos no consideran. Sin embargo, con el fin de visualizar algunas de las diferencias entre los resultados, se presentan los datos de promedios y desviaciones estándar de las tablas de resultados (Ver Tablas 10, 11 y 14). Esa información se presenta a continuación en la Tabla 16.

Tabla 16. Resultados de la programación con diferentes métodos.

	NUMERO VUELTAS	ESPERA TOTAL	JORNADA LABORAL	ESPERA MIN	ESPERA MAX	ESPERA PROMEDIO	TIEMPO/ VUELTA
Programación Clásica con tiempo de Terminal (P.C.T.T.)							
PROM	7,73	02:27	14:33	00:05	00:40	00:19	01:52
DES.EST	0,4403474	00:19	00:54	00:03	00:08	00:01	00:01
Programación Clásica sin tiempo de Terminal (P.S.T.T.)							
PROM	8,17	02:30	14:08	00:06	00:38	00:18	01:43
DES.EST	0,372678	00:23	01:15	00:04	00:04	00:02	00:05
Programación con Algoritmos Genéticos (P.A.G.)							
PROM	7,74	00:41	12:46	00:00	00:16	00:05	01:39
DES.EST	0,635949	00:19	00:58	00:00	00:06	00:02	00:02

La diferencia entre la programación con tiempo de Terminal (P.C.T.T.) y la programación sin tiempo de Terminal (P.S.T.T.), es que la primera incluye el tiempo de Terminal, que para la ruta piloto es de cinco (5) minutos adicionales al tiempo de recorrido y la segunda no. La programación con Algoritmos Genéticos (P.A.G.) incluye el tiempo de Terminal, porque aunque hace necesario utilizar un vehículo adicional, mantiene un margen en la programación que la hace menos susceptible a posibles retrasos o imprevistos.

Considerando los valores promedio, la programación obtenida mediante Algoritmos Genéticos domina a la programación clásica con tiempo de Terminal, es decir, es

mejor en todos los criterios. No es apropiado comparar la programación obtenida mediante Algoritmos Genéticos con la programación clásica sin tiempo de Terminal, debido a que esta última no considera el tiempo de holgura en Terminal de cinco (5) minutos, lo que hace que fácilmente supere a la primera programación en número de vueltas, debido a que cada vuelta es de menor duración. Sin embargo es notable que la obtenida por Algoritmos genéticos domina en todos los otros criterios a la segunda programación.

Obviamente, los resultados dan diferente. Los métodos clásicos sólo consideran un objetivo, en este caso minimizar el número de vehículos y se observa que el resultado obtenido para este objetivo es igualado por el algoritmo genético (siempre y cuando se mantengan los parámetros de diseño). Es muy difícil o casi imposible que una heurística (el método manual) sea capaz de encontrar un óptimo global en un problema con tantas posibilidades (combinatorial), y menos aun cuando se consideran múltiples criterios.

Si se compara la solución obtenida al optimizar un único criterio i con la obtenida al introducir otros criterios (como por ejemplo: número de vueltas, tiempos de espera, duración de la jornada laboral y programación de almuerzos y descansos), es de esperarse que, respecto al criterio i , la segunda solución sea peor que la primera, ya que si se quiere obtener mejores soluciones en los otros criterios, se requiere sacrificar en el criterio i . Esto debido a que los objetivos suelen ser conflictivos entre sí y no es posible encontrar una solución óptima para todos los criterios simultáneamente, sino muchas soluciones eficientes (pareto optimas: para encontrar una solución mejor en un criterio hay que sacrificar en otro), por lo que para definir cual de ellas implementar realmente, es necesario introducir valoraciones subjetivas que hacen imposible incluso que puedan compararse resultados ofrecidos por uno u otro analista. Se debe elegir una programación tal que, integrando todas las expectativas, sea justa, factible y eficiente y que satisfaga los diferentes criterios.

Lo que se muestra a continuación es la comparación de las programaciones en los cuatro criterios establecidos para el Algoritmo genético (número de rutinas, promedio de vueltas, tiempos de espera y programación de almuerzos y descansos).

Con respecto al número de rutinas, el menor número de vehículos se obtiene con la P.S.T.T. (18 vehículos), mientras que con el algoritmo genético y la P.C.T.T. se requieren 19 vehículos. Hay que tener en cuenta que en P.S.T.T. se está eliminando el Tiempo de Terminal, lo cual va violando los parámetros de diseño de la ruta y hace que la programación sea demasiado estricta porque no maneja un tiempo mínimo de descanso entre recorridos.

Al tener menor número de vehículos, la P.S.T.T. tiene el mayor promedio de vueltas, pero entre P.S.T.T. y P.A.G, el promedio es muy similar, aunque P.A.G. tiene mayor desviación estándar, indicando que los datos están más lejos del promedio. Pero este es parte del costo que se tiene que pagar por tratar de incluir otros criterios en la optimización.

La espera total promedio de P.A.G. con respecto a las demás es mucho menor (41 minutos comparado con 2:27 y 2:30), mostrando que respecto a este criterio el algoritmo genético ahorra gran cantidad de tiempo de espera entre recorridos. Este ahorro se ve reflejado además en la jornada laboral, que se reduce en promedio más de 1 hora y 20 minutos. Hay que tener en cuenta que para el algoritmo genético se programaron los almuerzos y descansos, quitándolos de las esperas y de la jornada laboral. Las programaciones manuales no tienen ni almuerzos ni descansos programados y los tiempos que se podrían tomar como almuerzo son a horas no apropiadas.

La espera mínima en el algoritmo genético es 0 minutos y la máxima es en promedio 16 minutos. Mientras que en las programaciones manuales son la mínima de 5 y 6 minutos y la máxima de 40 y 38 minutos, respectivamente. Esto indica que mientras en las programaciones normales hay descansos largos en cada uno de los despachos, en el Algoritmo Genético se reducen al máximo las esperas en cada despacho y se introduce una o dos esperas largas para tomar el almuerzo o para descansar.

Otra mejora del algoritmo genético se refleja en el tiempo promedio por vuelta, que para las programaciones manuales es de 1:52 y 1:43, mientras que para el Algoritmo Genético es de 1:39.

El Algoritmo genético es abierto a la hora de introducir nuevos elementos, por ejemplo uno podría buscar una solución imponiéndole la restricción de que el Terminal de despacho nunca se quede con menos de cierto número de vehículos en espera. Eso con el fin de hacer la programación menos vulnerable, también se pueden manejar otros parámetros como el máximo número de vueltas que puede tener un vehículo o el número máximo de vueltas antes de un descanso. Además se puede configurar el Algoritmo Genético para tratar de optimizar con un número fijo de vehículos, esto es difícil de hacer con la programación clásica, pero es deseable en vista de que las empresas tienen una gran sobreoferta de vehículos que las obliga a programar más vehículos de los que se obtienen con la programación óptima.

Tabla 17. Comparación de resultados utilizando diferentes criterios.

ALTERNATIVAS	Equidad en Número de Vueltas		Número rutinas	Tiempo de espera en Terminal		Almuerzos		
	No.prom. vueltas	max. Dif. Rutinas	No. Rutinas	espera total prom	Jornada promedio	% que almuerza	tiempo alm prom	% que almuerzo hora OK
Método manual	7,73	1	19	02:27	14:33			
AG con pesos iguales	7,73	2	19	00:41	12:46	100%	01:27	57.89%
AG considerando solo criterio 1	7,73	1	19	00:55	14:22	78.94%	00:43	26.31%
AG considerando solo criterio 2	7,73	3	19	00:48	14:25	94,73%	01:01	47.36%
AG considerando solo criterio 3	6,12	7	24	00:26	13:25	100%	01:53	33.33%
AG considerando solo criterio 4	6,39	5	23	01:11	13:34	100%	01:14	100%

Es de aclarar que para la solución generada por Algoritmos Genéticos se consideró pesos de importancia relativa iguales, pero a juicio del analista pueden introducirse otros valores diferentes, e incluso no considerar alguno de los criterios asignándole un peso de cero. Es importante además no implementar nunca la solución sin hacer una análisis de sensibilidad, esto es, analizar si hay fuertes cambios en la programación definida al considerar pequeños cambios en los pesos de importancia usados.

En la Tabla 17 se presentan los resultados comparando la programación clásica manual, el algoritmo genético con pesos iguales en los cuatro criterios y el algoritmo

genético considerando uno sólo de los criterios. El criterio 1 es la equidad en el número de vueltas, el criterio 2 es el mínimo número de vehículos o rutinas, el mínimo tiempo de espera en Terminal es el criterio 3 y el criterio 4 es el tiempo programado para almorzar a horas convenientes. En este último criterio se establecieron como horas convenientes las comprendidas entre las 11:30 a.m. y las 3:30 p.m. La última columna corresponde al porcentaje de rutinas que tienen almuerzo programado en ese intervalo de tiempo. (Ver Tabla 17).

Los datos de la Tabla 17, aparecen normalizados en la Tabla 18, con el objeto de poder compararlos entre sí. Los resultados de la comparación se pueden apreciar mejor en la Figura 8.

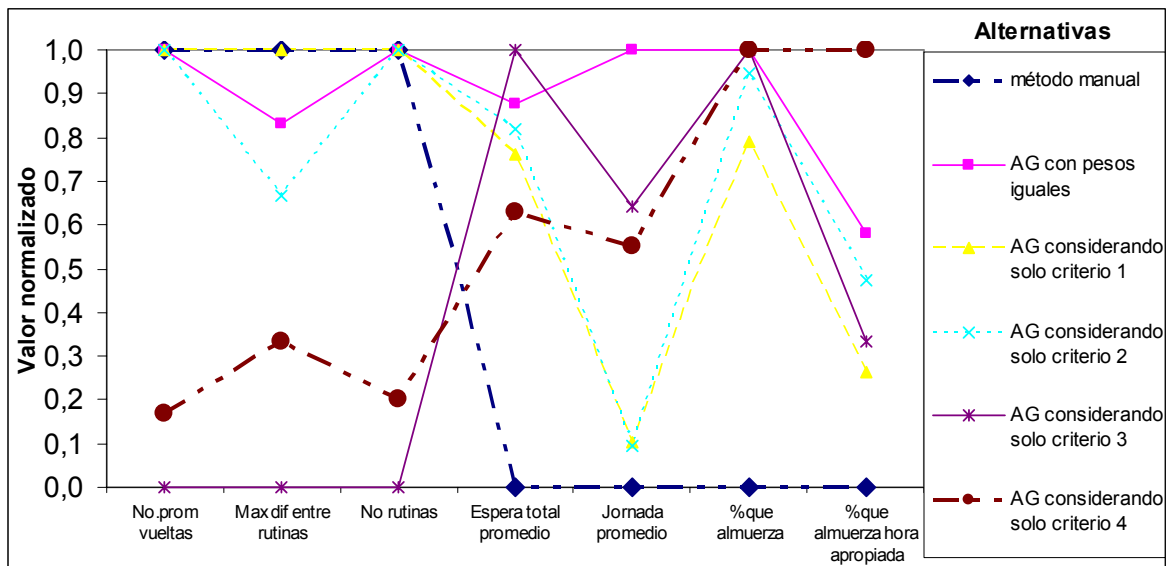
Tabla 18. Comparación de resultados utilizando diferentes criterios (Normalizada).

ALTERNATIVAS	No.prom. vueltas	max. Dif. Rutinas	No. Rutinas	espera total prom	Jornada promedio	% que almuerza	% que almuerza hora OK
Método manual	1,0	1,0	1,0	0,0	0,0	0,0	0,0
AG con pesos Iguales	1,0	0,8	1,0	0,9	1,0	1,0	0,6
AG considerando Solo criterio 1	1,0	1,0	1,0	0,8	0,1	0,8	0,3
AG considerando Solo criterio 2	1,0	0,7	1,0	0,8	0,1	0,9	0,5
AG considerando Solo criterio 3	0,0	0,0	0,0	1,0	0,6	1,0	0,3
AG considerando Solo criterio 4	0,2	0,3	0,2	0,6	0,6	1,0	1,0
	Max	Min	Min	Min	Min	Max	Max
Max	7,73	-1	-19	-26	-12,77	1	1
Min	6,12	-7	-24	-147	-14,55	0	0

De la Tabla 18 y la Figura 8 se puede concluir que si se comparan el método manual con la programación generada por el algoritmo genético, teniendo en cuenta únicamente el criterio 2, es decir, considerando únicamente el mínimo número de rutinas, que sería el criterio que utiliza la programación clásica, se nota que el Algoritmo Genético domina a la programación manual. Esto quiere decir que el Algoritmo Genético es mejor o igual en todos los criterios a la programación manual,

cuando únicamente se tiene en cuenta el mismo criterio de minimizar el número de vehículos que se utilizan en la programación. Las otras programaciones que consideran un único criterio son peores que la programación manual en algunos aspectos, pero muy superiores en otros criterios, como se esperaría que sucediera. Lo que es notable es que al comparar la programación clásica con el Algoritmo Genético que tiene en cuenta todos los criterios bajo la misma ponderación, el Algoritmo Genético iguala y supera a la programación manual clásica en todos los aspectos, excepto uno.

Figura 8. Comparación de las diferentes programaciones.



La incorporación de varios criterios considera otras ópticas diferentes al número de vehículos, que también son muy importantes, y que los métodos clásicos no consideran. Esto permite además una mejor planificación que redundará en beneficios para los actores que intervienen en el Transporte Público Urbano Colectivo.

Los usuarios, gracias a los itinerarios adecuados y a los intervalos que se mantienen, experimentarán la conveniencia y comodidad de un buen servicio, menor contaminación, mayor velocidad en sus desplazamientos y menor tiempo empleado en los viajes.

Los conductores disfrutarán de un mejor ambiente laboral, menos estrés y mayor equidad, ya que cuando se presentan deficiencias en la programación, las jornadas de trabajo son excesivamente largas y algunos conductores trabajan más que otros.

Las empresas y los propietarios también se beneficiarán puesto que la programación de rodamientos es la base de una operación ordenada y controlada. La operación no óptima da como resultado un servicio poco atractivo a los usuarios a la vez de reeditar un menor ingreso al propietario y al conductor.

La programación óptima también beneficiará a los organismos de control ya que al conocerse los tiempos en que los vehículos deben salir o llegar a un Terminal, es más fácil observar las desviaciones en estos tiempos y aplicar medidas correctivas que garanticen un buen servicio para los usuarios.

6.3 Resultados de la programación semanal.

Como ya se había mencionado, una vez obtenida la programación óptima con cierto número de rutinas, es necesario asignarlas diariamente a los diferentes vehículos de la empresa. Esta asignación semanal pretende que los vehículos de la empresa tengan una asignación equitativa de rutinas a lo largo de la semana, además se contemplan más vehículos de los necesarios para programar los descansos diarios.

Se probaron dos modelos: uno teniendo en cuenta la valoración de las diferentes rutinas denominado modelo con preferencias y otro sin tener en cuenta dicha valoración, llamado modelo sin preferencias. Los cálculos de esas preferencias aparecen en la Tabla 18.

6.2.1 Resultados del modelo sin preferencias:

Se supusieron 25 buses activos, aunque la programación solo requiere de 19 de ellos cada día. Eso quiere decir que cada día descansan 6 buses. La asignación quedaría así:

Máximo asignamiento 4

Mínimo asignamiento 3

Esto quiere decir que de los 5 días hábiles de la semana le toca trabajar a cada vehículo máximo 4 días y mínimo 3 días. Los resultados de la programación se observan en la Tabla 17.

Tabla 19. Programación semanal sin preferencias.

Rutina	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	Buses																		
Lunes	4	10	20	19	8	22	5	6	14	16	15	25	13	12	11	1	3	7	9
Martes	18	23	2	17	11	22	25	7	4	3	16	20	5	13	12	21	10	9	1
Miércoles	25	8	5	24	23	20	21	18	17	19	6	15	2	13	3	14	1	11	7
Jueves	22	15	9	4	23	21	20	19	18	17	24	14	6	7	12	11	2	8	5
Viernes	4	1	6	22	21	24	19	18	23	16	15	14	13	17	2	10	9	8	3

Los resultados de la Tabla 17 se interpretan de la siguiente manera: Al vehículo 4 le toca la siguiente programación: El día lunes la rutina número 1, el Martes la rutina 9, el miércoles descansa, el jueves la rutina 4 y el viernes repite la rutina 1. Y así sucesivamente con todos los vehículos.

6.2.2 Resultados del modelo con preferencias:

Este modelo incluye la valoración de preferencias presentada en la Tabla 18. Los resultados son los siguientes:

máximo asignamiento ponderado: 3.045

mínimo asignamiento ponderado: 1.834

La Tabla 19 muestra los resultados de la asignación. Esta tabla se interpreta como la anterior, por ejemplo: para el vehículo 1 la programación es: lunes rutina 10, martes rutina 6, miércoles descansa, jueves rutina 11, viernes rutina 12. Además según esta programación, el lunes descansan los vehículos 13, 14, 19, 21, 22, 24 y 25.

Tabla 20. Cálculo de las preferencias.

Rutina	Vueltas	Esperas	Jornada	Normalizada			Promedio
0	8	00:25:00	14:05:00	0,7	1,0	0,7	0,79
1	9	00:56:00	15:35:00	1,0	0,8	0,0	0,59
2	8	00:30:00	13:34:00	0,7	0,9	1,0	0,86
3	8	00:31:00	13:56:00	0,7	0,9	0,8	0,80
4	8	00:20:00	14:00:00	0,7	1,0	0,8	0,81
5	9	00:36:00	15:26:00	1,0	0,9	0,1	0,66
6	8	00:53:00	15:11:00	0,7	0,8	0,2	0,55
7	8	00:18:00	13:40:00	0,7	1,0	1,0	0,87
8	8	00:42:00	14:04:00	0,7	0,9	0,8	0,76
9	8	00:49:00	14:27:00	0,7	0,8	0,6	0,68
10	8	00:26:00	13:35:00	0,7	1,0	1,0	0,87
11	8	01:14:00	13:42:00	0,7	0,7	0,9	0,75
12	8	00:49:00	13:58:00	0,7	0,8	0,8	0,76
13	7	00:21:00	15:06:00	0,3	1,0	0,2	0,52
14	8	00:41:00	13:55:00	0,7	0,9	0,8	0,78
15	7	01:08:00	15:16:00	0,3	0,7	0,2	0,40
16	7	00:36:00	15:31:00	0,3	0,9	0,0	0,42
17	6	03:03:00	13:51:00	0,0	0,0	0,9	0,29
18	6	00:37:00	14:11:00	0,0	0,9	0,7	0,53

Tabla 21. Programación semanal con preferencias.

Rutina	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	Buses																		
Lunes	5	17	3	15	8	19	11	4	10	1	7	2	20	6	16	18	12	23	9
Martes	24	2	11	17	18	1	3	16	22	8	6	13	23	14	4	10	12	20	9
Miercoles	5	2	22	4	21	25	11	12	13	10	14	20	8	3	7	19	9	24	17
Jueves	23	9	22	7	16	21	5	19	6	18	1	25	10	15	14	4	12	2	13
Viernes	25	17	9	5	23	19	18	14	24	13	12	1	20	16	15	11	21	8	3

Estos modelos se optimizaron utilizando el software especializado GAMS. El código fuente de cada uno de los modelos se incluye en los Anexos 10 y 11.

CONCLUSIONES Y PERSPECTIVAS FUTURAS

El transporte público en Colombia tiene una estructura y un esquema de funcionamiento complejo, en el que intervienen diferentes actores con intereses particulares que pueden ser conflictivos entre sí. Esta situación hizo necesario el diseño e implementación de un modelo particular para la optimización de la programación de los vehículos.

Los trabajos en otros países, relacionados con el tema de la programación de itinerarios o rodamientos, no son aplicables a las condiciones del Transporte Público en Colombia.

Aunque la solución a todos los problemas del Transporte Público no está en la optimización de la programación, dicha optimización contribuye en gran manera a reducirlos y puede ser la alternativa más sencilla de implementar para mejorar las condiciones de operación del sistema.

La programación actual de rodamientos se hace de forma manual, sin tener en cuenta ningún criterio de optimización y de una forma deficiente. Esa programación de rodamientos e itinerarios no permite una operación eficiente y contribuye a empeorar los problemas del Transporte.

Una programación eficiente de rodamientos debe incluir varios objetivos, muchos de los cuales son conflictivos entre sí, pero todos son necesarios para que se pueda ofrecer un buen servicio a un costo de operación menor.

Las técnicas clásicas de optimización no son apropiadas para la solución de este problema debido a la alta dimensionalidad del mismo y a que algunas de las funciones

objetivo a usar son no lineales, algunas son imposibles de expresar en una función y otras solo se pueden expresar mediante funciones a tramos.

Las técnicas metaheurísticas son una buena aproximación para solucionar el problema, pero de ellas se escogieron los algoritmos genéticos porque se consideran la herramienta más conveniente, según lo reporta la literatura técnica, además facilitan el uso de funciones objetivo con las características descritas anteriormente y exploran simultáneamente muchas de las posibles situaciones, lo que no hacen las otras metaheurísticas.

En el problema del despacho de vehículos de transporte público hay una restricción física inviolable y es la de solo despachar vehículos que hayan concluido el despacho anterior. Los algoritmos genéticos permiten aplicar mecanismos de reparación, en lugar de definir todas las condiciones posibles, facilitando la generación de soluciones factibles sin complicar el modelo.

Los resultados de las pruebas del algoritmo genético, muestran que la programación obtenida mediante él no puede ser mejor a la programación clásica en todos los criterios, pero permite optimizar varios criterios en forma simultánea. Algo que está fuera del alcance del método convencional.

Mientras que en las programaciones clásicas solo se puede incluir un criterio para optimizar, el Algoritmo Genético ofrece flexibilidad para incluir varios criterios y para ponderar cada uno de ellos de acuerdo a las prioridades o intereses de la empresa de transporte. Esto además permite obtener soluciones rápidas y apropiadas sin repetir el proceso manual de programación.

TRABAJOS FUTUROS

Un trabajo futuro sería la integración de varias rutas en la programación, para que un vehículo no tenga que estar trabajando en la misma ruta todo el día, sino que pueda trabajar entre diferentes rutas que compartan el mismo despacho.

La programación eficiente de los rodamientos es solo una fase necesaria para que el sistema de transporte público funcione adecuadamente. Las otras dos fases son la operación y el control. Los trabajos futuros, que ya se están desarrollando, tienen que ver con el diseño e implementación de un sistema de despacho y control basado en tarjetas inteligentes, que facilite el cumplimiento de la programación y la aplicación de los correctivos necesarios. Al unir este programa de despacho y control con el algoritmo genético para la programación óptima de rodamientos, se logrará un sistema económico que realmente mejore el servicio de Transporte Público en las ciudades colombianas sin necesidad de reemplazar los vehículos actuales o de implementar costosos sistemas de transporte masivo.

Un artículo basado en la presente tesis de maestría fue aceptado y presentado en el Congreso Latino-Iberoamericano de Investigación de Operaciones (CLAIO 2004) realizado en Cuba del 4 al 8 de octubre de 2004. Algunos paper seleccionados del congreso se publicarán en un número especial de la revista ANNALS OF OPERATIONS RESEARCH y en la Revista JOURNAL OF HEURISTICS.

REFERENCIAS

- BEASLEY, J.E. CHU, P.C (1994). A Genetic Algorithm for the Set Covering Problem. Londres: Management School, 1994. 19 pags.
- BODIN, L.D., Ball. M.O. y GREENBERG, J. (1995). Enhancements to the RUCUS II crew scheduling system, in Rousseau, J.M. (Ed.), Computer Aided Scheduling of Public Transport 2, North-Holand.
- BORNDÖRFER, R. et al. (1998). Optimization of Transportation Systems. ZIB Report 98-09. Zuse Institute Berlin (ZIB).
- BORNDÖRFER, R. et al. (2003). Vehicle & Duty Scheduling in Public Transit. Zuse Institute Berlin (ZIB).
- CAPRARA, A. (1995). Modelling and solving the crew rostering problem. Universidad de Bologna. Technical Report DEIS 95-6.
- COELLO, C. A. (2002) Evolutionary Algorithms for Solving Multi-Objective Problems. New York: Kluwer Academic.
- CURTIS, S.D. (2000). Constraint Satisfaction Approaches to Bus Driver Scheduling. University of Leeds, School of Computing, PhD. Thesis.
- DIAS, T. G. SOUSA, J. P., CUNHA, J. (2001) A genetic Algorithm for the Bus Driver Scheduling Problem. En: MIC'2001 – 4th Metaheuristics International Conference. Porto, Julio 16-20, 2001.
- FANG, H. (1994) Genetics algorithms in timetabling and scheduling. University of Edinburgh, PhD. Tesis.
- FORES, S. (1996) "Column generation approaches to bus driver scheduling." University of Leeds PhD Thesis.
- GLOVER, F. LAGUNA, M. (1997). Tabu Search, Kluwer Academic Publishers.
- GOICOECHEA A., HANSEN, D. y DUCKSTEIN, L. (1982). Multiobjective decision analysis with engineering and business applications. John Wiley and Sons.
- GROBNER, M. (1998). Rostering with a hibrid Genetic Algorithm.

GRUTTNER, M. A. (2001). Algoritmos Genéticos en Recorridos Optimos de Líneas de Transporte Público. Chile: Universidad de Concepción.

HUISMAN, D. WAGELMANS, A. (2004). A solution approach for dynamic vehicle and crew scheduling. Rotterdam: Erasmus Center for Optimization in Public Transport (ECOPT) & Econometric Institute, Erasmus University Rotterdam.

JASZKIEWICZ, A. (1996). Methods for solving multiple objective combinatorial problems. Documento suministrado directamente por el autor.

KOCJAN, W. (2001). Heuristic Methods for routing and scheduling. SICS Technical Report. T2001:17.

LA PATRIA (2004). Denuncia por contaminación producida por Buses en la Carrera 21. En: Periódico La Patria.. Agosto 26 de 2004.

La Vial Ingenieros Ltda. (2001). Actualización de Variables de Transporte Público. Informe Final. Mayo de 2001.

LI, J. (2002). Fuzzy Evolutionary Approaches for Bus and Rail Driver Scheduling. University of Leeds PhD. Thesis.

LOURENCO, H. et al. (2000). Metaheuristics for the Bus-driver Scheduling Problem. Barcelona: Department of Economics and Management. Universitat Pompeu Fabra.

MACHADO, P. et al. (2000). Vehicle Routing Problem: Doing it the Evolutionary Way. Portugal: Instituto Superior de Engenharia de Coimbra.

MAUTTONE, A. et al. (2004). Diseño y Optimización de Rutas y Frecuencias en el Transporte Colectivo Urbano, Modelos y Algoritmos. Universidad de la República Uruguay. Tesis de Maestría.

MELO G., L. (2004). Los pecados del Transporte Público. En: Periódico El País, Colombia, mayo 14 de 2001, p. 1B.

MOLINERO M., A. SANCHEZ A., I. (1998). Transporte Público: Planeación, Diseño, Operación y Administración. 3ª Edición. México: Fundación ICA.

MOLLER, R. (2001) Una Propuesta de Mejoramiento del Transporte Público Colectivo Urbano en Santiago de Cali. En: Ingeniería y Competitividad. Universidad del valle. Volumen 3. No.1. Junio de 2001. Pags. 19-34.

PINE, R. et al. (1998). TCRP Report 30. Transit Scheduling: Basic and Advanced Manuals. Washington: Transportation Research Board.

REVISTA CAMBIO (2004). Primera Estación. En: Revista Cambio, Agosto de 2004, Colombia.

ROMERO, C. (1993). Teoría de la decisión multicriterio, conceptos, técnicas y aplicaciones. Alianza editorial.

SCHAFFER, J. (1994). Some experiments in mashing learning using vector evaluated genetic algorithms. Ph. D. dissertation, Vanderbilt University, Nashville.

SERAFINI, P. (1992) Simulated annealing for multi objective optimization problems, *Proceedings of the Xth International Conference on MCDM*, pp 87-96.

SHEN, Y. KWAN, R.S.K. (2001). Tabu search for driver scheduling. In VoB, S. y Daduna, J.R. (Eds.) *Computer Aided Scheduling of Public Transport*, Springer-Verlag.

SMITH-KEARY, P. (1979). *Genetic, Structure and Function*, Macmillan Press.

SMITH, R., MESA O., DYNER I., JARAMILLO P., POVEDA G., VALENCIA D. (2000). *Decisiones con múltiples objetivos e incertidumbre*. 2ª edición. Universidad Nacional de Colombia. Facultad de Minas, Posgrado Recursos Hidráulicos. Medellín.

SRIVINAS, N. y KALYANMOY, D. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Proceedings of IEEE World Congress on Computational Intelligence*, Orlando.

TAHA, H. (1997). *Operations Research (An Introduction)*. Prentice Hall.

WREN, A. SMITH, B.M. (1998). Experiences with a crew scheduling system based on set covering. In DADUNA, J.R. y WREN, A (Eds.) *Proceedings of the fourth International Workshop on Computer-Aided Scheduling of Public Transport*, Springer-Verlag.

WREN, A. (1997). *Integrated scheduling of buses and drivers*. University of Leeds SCHOOL OF COMPUTER STUDIES, RESEARCH REPORT SERIES. Report 97.32.

ANEXOS

ANEXO 1. Programación clásica manual con Tiempo de Terminal.

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
	05:40	07:11	1	00:14
	05:48	07:19	2	00:11
	05:56	07:27	3	00:08
	06:04	07:35	4	00:05
	06:10	07:45	5	00:02
	06:15	07:50	6	00:04
	06:20	07:55	7	00:06
	06:25	08:00	8	00:08
	06:30	08:05	9	00:10
	06:35	08:10	10	00:12
	06:40	08:15	11	00:14
	06:45	08:20	12	00:16
	06:50	08:25	13	00:18
	06:55	08:30	14	00:20
	07:00	08:35	15	00:22
	07:05	08:40	16	00:24
	07:10	08:45	17	00:26
1	07:15	08:50	18	00:28
1,2	07:20	08:55	19	00:30
2,3	07:25	09:00	1	00:32
3	07:30	09:05	2	00:34
4	07:35	09:10	3	00:36
	07:40	09:13	4	00:40
	07:47	09:20	5	00:40
	07:54	09:27	6	00:40
8	08:01	09:34	7	00:40
9	08:08	09:41	8	00:40
10,11	08:15	09:48	9	00:40
11,12	08:22	09:55	10	00:40
12,13	08:29	10:02	11	00:40
13,14,15	08:36	10:09	12	00:40
14,15,16	08:43	10:16	13	00:40
15,16,17,18	08:50	10:23	14	00:37
16,17,18,19	08:57	10:30	15	00:35
17,18,19,1	09:04	10:37	16	00:33
18,19,1,2,3	09:11	10:44	17	00:31
19,1,2,3,4	09:18	10:51	18	00:29
1,2,3,4,5	09:25	10:58	19	00:27
2,3,4,5,6	09:32	11:05	1	00:25

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
3,4,5,6,7	09:39	11:12	2	00:23
4,5,6,7,8	09:46	11:19	3	00:21
5,6,7,8,9	09:53	11:26	4	00:19
6,7,8,9,10	10:00	11:33	5	00:17
7,8,9,10,11	10:07	11:40	6	00:15
8,9,10,11,12	10:14	11:47	7	00:13
9,10,11,12,13	10:21	11:54	8	00:13
10,11,12,13,14	10:28	12:01	9	00:13
11,12,13,14,15	10:35	12:08	10	00:13
12,13,14,15,16	10:42	12:15	11	00:13
13,14,15,16,17	10:49	12:22	12	00:13
14,15,16,17,18	10:56	12:29	13	00:13
15,16,17,18,19	11:00	12:35	14	00:14
16,17,18,19,1	11:05	12:40	15	00:15
17,18,19,1	11:10	12:45	16	00:15
18,19,1,2	11:15	12:50	17	00:15
19,1,2,3	11:20	12:55	18	00:15
1,2,3	11:25	13:00	19	00:15
2,3,4	11:30	13:05	1	00:15
3,4,5	11:35	13:10	2	00:15
4,5,6	11:40	13:15	3	00:15
5,6	11:45	13:20	4	00:15
6,7	11:50	13:25	5	00:15
7,8	11:55	13:28	6	00:17
8	12:00	13:33	7	00:17
9	12:07	13:40	8	00:15
10	12:14	13:47	9	00:13
11	12:21	13:54	10	00:13
12	12:28	14:01	11	00:13
13,14	12:35	14:08	12	00:13
14,15	12:42	14:15	13	00:13
15,16	12:49	14:22	14	00:13
16,17,18	12:55	14:30	15	00:12
17,18,19	13:00	14:35	16	00:14
18,19,1	13:05	14:40	17	00:16
19,1,2	13:10	14:45	18	00:18
1,2,3	13:15	14:50	19	00:20
2,3,4	13:20	14:55	1	00:22
3,4,5	13:25	15:00	2	00:24

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
4,5,6	13:30	15:05	3	00:26
5,6,7	13:35	15:10	4	00:28
6,7,8	13:40	15:15	5	00:30
7,8	13:45	15:20	6	00:32
8,9	13:50	15:25	7	00:34
9,10	13:55	15:30	8	00:36
10	14:00	15:33	9	00:40
11	14:07	15:40	10	00:40
12	14:14	15:47	11	00:40
13	14:21	15:54	12	00:40
14	14:28	16:01	13	00:40
15,16	14:35	16:08	14	00:40
16,17	14:42	16:15	15	00:40
17,18	14:49	16:22	16	00:38
18,19,1	14:56	16:29	17	00:36
19,1,2	15:03	16:36	18	00:34
1,2,3,4	15:10	16:43	19	00:32
2,3,4,5	15:17	16:50	1	00:30
3,4,5,6	15:24	16:57	2	00:28
4,5,6,7,8	15:31	17:04	3	00:26
5,6,7,8,9	15:38	17:11	4	00:24
6,7,8,9,10	15:45	17:18	5	00:22
7,8,9,10,11	15:52	17:25	6	00:20
8,9,10,11,12	15:59	17:32	7	00:18
9,10,11,12,13	16:06	17:39	8	00:16
10,11,12,13,14	16:13	17:46	9	00:14
11,12,13,14,15	16:20	17:53	10	00:12
12,13,14,15,16	16:27	18:00	11	00:10
13,14,15,16,17	16:34	18:07	12	00:08
14,15,16,17,18	16:41	18:14	13	00:06
15,16,17,18,19	16:48	18:21	14	00:04
16,17,18,19,1	16:55	18:28	15	00:02
17,18,19,1,2	17:00	18:35	16	00:00
18,19,1,2,3	17:05	18:40	17	00:00
19,1,2,3	17:10	18:45	18	00:02
1,2,3,4	17:15	18:50	19	00:04
2,3,4,5	17:20	18:55	1	00:06
3,4,5,6	17:25	19:00	2	00:08
4,5,6	17:30	19:05	3	00:10
5,6,7	17:35	19:10	4	00:12
6,7,8	17:40	19:15	5	00:14
7,8	17:45	19:20	6	00:16
8,9	17:50	19:25	7	00:18
9,10	17:55	19:30	8	00:20
10,11	18:00	19:35	9	00:25
11	18:05	19:40	10	00:30
12	18:10	19:45	11	00:35

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
13	18:15	19:50	12	00:40
14	18:20	19:55	13	00:55
	18:25	20:00	14	01:10
	18:30	20:05	15	
	18:35	20:10	16	
	18:40	20:15	17	
	18:47	20:20	18	
	18:54	20:27	19	
2	19:01	20:34	1	
3	19:08	20:41	2	
4,5	19:15	20:48	3	
5,6	19:22	20:55	4	
6,7	19:29	21:02	5	
7,8,9	19:36	21:09	6	
8,9,10	19:43	21:16	7	
9,10,11,12	19:50	21:23	8	
10,11,12,13,14	20:00	21:31	9	
11,12,13,14	20:10	21:41	10	
12,13,14	20:20	21:51	11	
13,14	20:30	22:01	12	
14	20:50	22:21	13	
	21:10	22:41	14	

ANEXO 2. Programación manual con Tiempo de Terminal ordenada por rutinas.

Hora Despacho	Hora Disponible	Rutina	Espera
05:40	07:11	1	00:14
07:25	09:00	1	00:32
09:32	11:05	1	00:25
11:30	13:05	1	00:15
13:20	14:55	1	00:22
15:17	16:50	1	00:30
17:20	18:55	1	00:06
19:01	20:34	1	
		8	02:24
05:48	07:19	2	00:11
07:30	09:05	2	00:34
09:39	11:12	2	00:23
11:35	13:10	2	00:15
13:25	15:00	2	00:24
15:24	16:57	2	00:28
17:25	19:00	2	00:08
19:08	20:41	2	
		8	02:23
05:56	07:27	3	00:08
07:35	09:10	3	00:36
09:46	11:19	3	00:21
11:40	13:15	3	00:15
13:30	15:05	3	00:26
15:31	17:04	3	00:26
17:30	19:05	3	00:10
19:15	20:48	3	
		8	02:22
06:04	07:35	4	00:05
07:40	09:13	4	00:40
09:53	11:26	4	00:19
11:45	13:20	4	00:15
13:35	15:10	4	00:28
15:38	17:11	4	00:24
17:35	19:10	4	00:12
19:22	20:55	4	
		8	02:23
06:10	07:45	5	00:02
07:47	09:20	5	00:40
10:00	11:33	5	00:17
11:50	13:25	5	00:15
13:40	15:15	5	00:30

Hora Despacho	Hora Disponible	Rutina	Espera
15:45	17:18	5	00:22
17:40	19:15	5	00:14
19:29	21:02	5	
		8	02:20
06:15	07:50	6	00:04
07:54	09:27	6	00:40
10:07	11:40	6	00:15
11:55	13:28	6	00:17
13:45	15:20	6	00:32
15:52	17:25	6	00:20
17:45	19:20	6	00:16
19:36	21:09	6	
		8	02:24
06:20	07:55	7	00:06
08:01	09:34	7	00:40
10:14	11:47	7	00:13
12:00	13:33	7	00:17
13:50	15:25	7	00:34
15:59	17:32	7	00:18
17:50	19:25	7	00:18
19:43	21:16	7	
		8	02:26
06:25	08:00	8	00:08
08:08	09:41	8	00:40
10:21	11:54	8	00:13
12:07	13:40	8	00:15
13:55	15:30	8	00:36
16:06	17:39	8	00:16
17:55	19:30	8	00:20
19:50	21:23	8	
		8	02:28
06:30	08:05	9	00:10
08:15	09:48	9	00:40
10:28	12:01	9	00:13
12:14	13:47	9	00:13
14:00	15:33	9	00:40
16:13	17:46	9	00:14
18:00	19:35	9	00:25
20:00	21:31	9	
		8	02:35

Hora Despacho	Hora Disponible	Rutina	Espera
06:35	08:10	10	00:12
08:22	09:55	10	00:40
10:35	12:08	10	00:13
12:21	13:54	10	00:13
14:07	15:40	10	00:40
16:20	17:53	10	00:12
18:05	19:40	10	00:30
20:10	21:41	10	
		8	02:40
06:40	08:15	11	00:14
08:29	10:02	11	00:40
10:42	12:15	11	00:13
12:28	14:01	11	00:13
14:14	15:47	11	00:40
16:27	18:00	11	00:10
18:10	19:45	11	00:35
20:20	21:51	11	
		8	02:45
06:45	08:20	12	00:16
08:36	10:09	12	00:40
10:49	12:22	12	00:13
12:35	14:08	12	00:13
14:21	15:54	12	00:40
16:34	18:07	12	00:08
18:15	19:50	12	00:40
20:30	22:01	12	
		8	02:50
06:50	08:25	13	00:18
08:43	10:16	13	00:40
10:56	12:29	13	00:13
12:42	14:15	13	00:13
14:28	16:01	13	00:40
16:41	18:14	13	00:06
18:20	19:55	13	00:55
20:50	22:21	13	
		8	03:05
06:55	08:30	14	00:20
08:50	10:23	14	00:37
11:00	12:35	14	00:14
12:49	14:22	14	00:13
14:35	16:08	14	00:40
16:48	18:21	14	00:04
18:25	20:00	14	01:10

Hora Despacho	Hora Disponible	Rutina	Espera
21:10	22:41	14	
		8	03:18
07:00	08:35	15	00:22
08:57	10:30	15	00:35
11:05	12:40	15	00:15
12:55	14:30	15	00:12
14:42	16:15	15	00:40
16:55	18:28	15	00:02
18:30	20:05	15	
		7	02:06
07:05	08:40	16	00:24
09:04	10:37	16	00:33
11:10	12:45	16	00:15
13:00	14:35	16	00:14
14:49	16:22	16	00:38
17:00	18:35	16	00:00
18:35	20:10	16	
		7	02:04
07:10	08:45	17	00:26
09:11	10:44	17	00:31
11:15	12:50	17	00:15
13:05	14:40	17	00:16
14:56	16:29	17	00:36
17:05	18:40	17	00:00
18:40	20:15	17	
		7	02:04
07:15	08:50	18	00:28
09:18	10:51	18	00:29
11:20	12:55	18	00:15
13:10	14:45	18	00:18
15:03	16:36	18	00:34
17:10	18:45	18	00:02
18:47	20:20	18	
		7	02:06
07:20	08:55	19	00:30
09:25	10:58	19	00:27
11:25	13:00	19	00:15
13:15	14:50	19	00:20
15:10	16:43	19	00:32
17:15	18:50	19	00:04
18:54	20:27	19	
		7	02:08

ANEXO 3. Programación clásica manual sin tiempo de Terminal.

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
	05:40	07:06	1	00:14
	05:48	07:14	2	00:11
	05:56	07:22	3	00:08
	06:04	07:30	4	00:05
	06:10	07:40	5	00:00
	06:15	07:45	6	00:02
	06:20	07:50	7	00:04
	06:25	07:55	8	00:06
	06:30	08:00	9	00:08
	06:35	08:05	10	00:10
	06:40	08:10	11	00:12
	06:45	08:15	12	00:14
	06:50	08:20	13	00:16
	06:55	08:25	14	00:18
	07:00	08:30	15	00:20
	07:05	08:35	16	00:22
1	07:10	08:40	17	00:24
1,2	07:15	08:45	18	00:26
2	07:20	08:50	1	00:28
3	07:25	08:55	2	00:30
4	07:30	09:00	3	00:32
	07:35	09:05	4	00:34
	07:40	09:08	5	00:38
	07:47	09:15	6	00:38
	07:54	09:22	7	00:38
9	08:01	09:29	8	00:38
10	08:08	09:36	9	00:38
11,12	08:15	09:43	10	00:38
12,13	08:22	09:50	11	00:38
13,14	08:29	09:57	12	00:38
14,15,16	08:36	10:04	13	00:38
15,16,17	08:43	10:11	14	00:38
16,17,18,1	08:50	10:18	15	00:38
17,18,1,2	08:57	10:25	16	00:35
18,1,2,3	09:04	10:32	17	00:33
1,2,3,4,5	09:11	10:39	18	00:31
2,3,4,5,6	09:18	10:46	1	00:29
3,4,5,6,7	09:25	10:53	2	00:27
4,5,6,7,8	09:32	11:00	3	00:25
5,6,7,8,9	09:39	11:07	4	00:23
6,7,8,9,10	09:46	11:14	5	00:21
7,8,9,10,11	09:53	11:21	6	00:19
8,9,10,11,12	10:00	11:28	7	00:17
9,10,11,12,13	10:07	11:35	8	00:15

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
10,11,12,13,14	10:14	11:42	9	00:13
11,12,13,14,15	10:21	11:49	10	00:11
12,13,14,15,16	10:28	11:56	11	00:11
13,14,15,16,17	10:35	12:03	12	00:11
14,15,16,17,18	10:42	12:10	13	00:11
15,16,17,18,1	10:49	12:17	14	00:11
16,17,18,1,2	10:56	12:24	15	00:11
17,18,1,2,3	11:00	12:30	16	00:12
18,1,2,3	11:05	12:35	17	00:14
1,2,3,4	11:10	12:40	18	00:15
2,3,4,5	11:15	12:45	1	00:15
3,4,5	11:20	12:50	2	00:15
4,5,6	11:25	12:55	3	00:15
5,6,7	11:30	13:00	4	00:15
6,7,8	11:35	13:05	5	00:15
7,8	11:40	13:10	6	00:15
8,9	11:45	13:15	7	00:15
9,10	11:50	13:20	8	00:15
10	11:55	13:23	9	00:17
11	12:00	13:28	10	00:17
12	12:07	13:35	11	00:15
13	12:14	13:42	12	00:13
14	12:21	13:49	13	00:11
15	12:28	13:56	14	00:11
16,17	12:35	14:03	15	00:11
17,18	12:42	14:10	16	00:11
18,1	12:49	14:17	17	00:11
1,2,3	12:55	14:25	18	00:10
2,3,4	13:00	14:30	1	00:12
3,4,5	13:05	14:35	2	00:14
4,5,6	13:10	14:40	3	00:16
5,6,7	13:15	14:45	4	00:18
6,7,8	13:20	14:50	5	00:20
7,8,9	13:25	14:55	6	00:22
8,9,10	13:30	15:00	7	00:24
9,10,11	13:35	15:05	8	00:26
10,11	13:40	15:10	9	00:28
11,12	13:45	15:15	10	00:30
12,13	13:50	15:20	11	00:32
13	13:55	15:25	12	00:34
14	14:00	15:28	13	00:38
15	14:07	15:35	14	00:38
16	14:14	15:42	15	00:38
17	14:21	15:49	16	00:38

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
18	14:28	15:56	17	00:38
1,2	14:35	16:03	18	00:38
2,3	14:42	16:10	1	00:38
3,4	14:49	16:17	2	00:38
4,5,6	14:56	16:24	3	00:36
5,6,7	15:03	16:31	4	00:34
6,7,8,9	15:10	16:38	5	00:32
7,8,9,10	15:17	16:45	6	00:30
8,9,10,11	15:24	16:52	7	00:28
9,10,11,12,13	15:31	16:59	8	00:26
10,11,12,13,14	15:38	17:06	9	00:24
11,12,13,14,15	15:45	17:13	10	00:22
12,13,14,15,16	15:52	17:20	11	00:20
13,14,15,16,17	15:59	17:27	12	00:18
14,15,16,17,18	16:06	17:34	13	00:16
15,16,17,18,1	16:13	17:41	14	00:14
16,17,18,1,2	16:20	17:48	15	00:12
17,18,1,2,3	16:27	17:55	16	00:10
18,1,2,3,4	16:34	18:02	17	00:08
1,2,3,4,5	16:41	18:09	18	00:06
2,3,4,5,6	16:48	18:16	1	00:04
3,4,5,6,7	16:55	18:23	2	00:02
4,5,6,7,8	17:00	18:30	3	00:00
5,6,7,8	17:05	18:35	4	00:00
6,7,8,9	17:10	18:40	5	00:00
7,8,9,10	17:15	18:45	6	00:02
8,9,10,11	17:20	18:50	7	00:04
9,10,11	17:25	18:55	8	00:06
10,11,12	17:30	19:00	9	00:08
11,12,13	17:35	19:05	10	00:10
12,13	17:40	19:10	11	00:12
13,14	17:45	19:15	12	00:14
14,15	17:50	19:20	13	00:16
15,16	17:55	19:25	14	00:18
16	18:00	19:30	15	00:20
17	18:05	19:35	16	00:25
18	18:10	19:40	17	00:30
	18:15	19:45	18	00:35
	18:20	19:50	1	00:40
	18:25	19:55	2	00:55
	18:30	20:00	3	01:10
	18:35	20:05	4	
	18:40	20:10	5	
	18:47	20:15	6	
	18:54	20:22	7	
9	19:01	20:29	8	
10	19:08	20:36	9	

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
11,12	19:15	20:43	10	
12,13	19:22	20:50	11	
13,14	19:29	20:57	12	
14,15,16	19:36	21:04	13	
15,16,17	19:43	21:11	14	
16,17,18,1	19:50	21:18	15	
17,18,1,2,3	20:00	21:26	16	
18,1,2,3	20:10	21:36	17	
1,2,3	20:20	21:46	18	
2,3	20:30	21:56	1	
3	20:50	22:16	2	
	21:10	22:36	3	

ANEXO 4. Programación clásica manual sin tiempo de Terminal ordenada por rutinas.

Hora Despacho	Hora Disponible	Rutina	Espera
05:40	07:06	1	00:14
07:20	08:50	1	00:28
09:18	10:46	1	00:40
11:15	12:45	1	00:15
13:00	14:30	1	00:12
14:42	16:10	1	00:38
16:48	18:16	1	00:04
18:20	19:50	1	00:40
20:30	21:56	1	
		9	03:11
05:48	07:14	2	00:11
07:25	08:55	2	00:30
09:25	10:53	2	00:27
11:20	12:50	2	00:15
13:05	14:35	2	00:14
14:49	16:17	2	00:38
16:55	18:23	2	00:02
18:25	19:55	2	00:55
20:50	22:16	2	
		9	03:12
05:56	07:22	3	00:08
07:30	09:00	3	00:32
09:32	11:00	3	00:25
11:25	12:55	3	00:15
13:10	14:40	3	00:16
14:56	16:24	3	00:36
17:00	18:30	3	00:00
18:30	20:00	3	01:10
21:10	22:36	3	
		9	03:22
06:04	07:30	4	00:05
07:35	09:05	4	00:34
09:39	11:07	4	00:23
11:30	13:00	4	00:15
13:15	14:45	4	00:18
15:03	16:31	4	00:34
17:05	18:35	4	00:00
18:35	20:05	4	
		8	02:09
06:10	07:40	5	00:00
07:40	09:08	5	00:38

Hora Despacho	Hora Disponible	Rutina	Espera
09:46	11:14	5	00:21
11:35	13:05	5	00:15
13:20	14:50	5	00:20
15:10	16:38	5	00:32
17:10	18:40	5	00:00
18:40	20:10	5	
		8	02:06
06:15	07:45	6	00:02
07:47	09:15	6	00:38
09:53	11:21	6	00:19
11:40	13:10	6	00:15
13:25	14:55	6	00:22
15:17	16:45	6	00:30
17:15	18:45	6	00:02
18:47	20:15	6	
		8	02:08
06:20	07:50	7	00:04
07:54	09:22	7	00:38
10:00	11:28	7	00:17
11:45	13:15	7	00:15
13:30	15:00	7	00:24
15:24	16:52	7	00:28
17:20	18:50	7	00:04
18:54	20:22	7	
		8	02:10
06:25	07:55	8	00:06
08:01	09:29	8	00:38
10:07	11:35	8	00:15
11:50	13:20	8	00:15
13:35	15:05	8	00:26
15:31	16:59	8	00:26
17:25	18:55	8	00:06
19:01	20:29	8	
		8	02:12
06:30	08:00	9	00:08
08:08	09:36	9	00:38
10:14	11:42	9	00:13
11:55	13:23	9	00:17
13:40	15:10	9	00:28
15:38	17:06	9	00:24
17:30	19:00	9	00:08

Hora Despacho	Hora Disponible	Rutina	Espera
19:08	20:36	9	
		8	02:16
06:35	08:05	10	00:10
08:15	09:43	10	00:38
10:21	11:49	10	00:11
12:00	13:28	10	00:17
13:45	15:15	10	00:30
15:45	17:13	10	00:22
17:35	19:05	10	00:10
19:15	20:43	10	
		8	02:18
06:40	08:10	11	00:12
08:22	09:50	11	00:38
10:28	11:56	11	00:11
12:07	13:35	11	00:15
13:50	15:20	11	00:32
15:52	17:20	11	00:20
17:40	19:10	11	00:12
19:22	20:50	11	
		8	02:20
06:45	08:15	12	00:14
08:29	09:57	12	00:38
10:35	12:03	12	00:11
12:14	13:42	12	00:13
13:55	15:25	12	00:34
15:59	17:27	12	00:18
17:45	19:15	12	00:14
19:29	20:57	12	
		8	02:22
06:50	08:20	13	00:16
08:36	10:04	13	00:38
10:42	12:10	13	00:11
12:21	13:49	13	00:11
14:00	15:28	13	00:38
16:06	17:34	13	00:16
17:50	19:20	13	00:16
19:36	21:04	13	
		8	02:26
06:55	08:25	14	00:18
08:43	10:11	14	00:38
10:49	12:17	14	00:11
12:28	13:56	14	00:11
14:07	15:35	14	00:38

Hora Despacho	Hora Disponible	Rutina	Espera
16:13	17:41	14	00:14
17:55	19:25	14	00:18
19:43	21:11	14	
		8	02:28
07:00	08:30	15	00:20
08:50	10:18	15	00:38
10:56	12:24	15	00:11
12:35	14:03	15	00:11
14:14	15:42	15	00:38
16:20	17:48	15	00:12
18:00	19:30	15	00:20
19:50	21:18	15	
		8	02:30
07:05	08:35	16	00:22
08:57	10:25	16	00:35
11:00	12:30	16	00:12
12:42	14:10	16	00:11
14:21	15:49	16	00:38
16:27	17:55	16	00:10
18:05	19:35	16	00:25
20:00	21:26	16	
		8	02:33
07:10	08:40	17	00:24
09:04	10:32	17	00:33
11:05	12:35	17	00:14
12:49	14:17	17	00:11
14:28	15:56	17	00:38
16:34	18:02	17	00:08
18:10	19:40	17	00:30
20:10	21:36	17	
		8	02:38
07:15	08:45	18	00:26
09:11	10:39	18	00:31
11:10	12:40	18	00:15
12:55	14:25	18	00:10
14:35	16:03	18	00:38
16:41	18:09	18	00:06
18:15	19:45	18	00:35
20:20	21:46	18	
		8	02:41

ANEXO 5. Programación clásica manual con 22 vehículos.

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
	05:40	07:06	1	00:34
	05:48	07:14	2	00:33
	05:56	07:22	3	00:32
	06:04	07:30	4	00:31
	06:10	07:40	5	00:28
	06:15	07:45	6	00:30
	06:20	07:50	7	00:32
	06:25	07:55	8	00:34
	06:30	08:00	9	00:36
	06:35	08:05	10	00:38
	06:40	08:10	11	00:40
	06:45	08:15	12	00:42
	06:50	08:20	13	00:44
	06:55	08:25	14	00:46
	07:00	08:30	15	00:48
	07:05	08:35	16	00:50
1	07:10	08:40	17	00:52
1,2	07:15	08:45	18	00:54
2	07:20	08:50	19	00:56
3	07:25	08:55	20	00:58
4	07:30	09:00	21	01:00
	07:35	09:05	22	01:02
	07:40	09:08	1	01:06
	07:47	09:15	2	01:06
	07:54	09:22	3	01:06
9	08:01	09:29	4	01:06
10	08:08	09:36	5	01:06
11,12	08:15	09:43	6	01:06
12,13	08:22	09:50	7	01:06
13,14	08:29	09:57	8	01:03
14,15,16	08:36	10:04	9	01:01
15,16,17	08:43	10:11	10	00:59
16,17,18,1	08:50	10:18	11	00:57
17,18,1,2	08:57	10:25	12	00:55
18,1,2,3	09:04	10:32	13	00:53
1,2,3,4,5	09:11	10:39	14	00:51
2,3,4,5,6	09:18	10:46	15	00:49
3,4,5,6,7	09:25	10:53	16	00:47
4,5,6,7,8	09:32	11:00	17	00:45
5,6,7,8,9	09:39	11:07	18	00:43
6,7,8,9,10	09:46	11:14	19	00:41
7,8,9,10,11	09:53	11:21	20	00:39
8,9,10,11,12	10:00	11:28	21	00:39
9,10,11,12,13	10:07	11:35	22	00:39
10,11,12,13,14	10:14	11:42	1	00:39

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
11,12,13,14,15	10:21	11:49	2	00:39
12,13,14,15,16	10:28	11:56	3	00:39
13,14,15,16,17	10:35	12:03	4	00:39
14,15,16,17,18	10:42	12:10	5	00:39
15,16,17,18,1	10:49	12:17	6	00:38
16,17,18,1,2	10:56	12:24	7	00:36
17,18,1,2,3	11:00	12:30	8	00:35
18,1,2,3	11:05	12:35	9	00:35
1,2,3,4	11:10	12:40	10	00:35
2,3,4,5	11:15	12:45	11	00:35
3,4,5	11:20	12:50	12	00:35
4,5,6	11:25	12:55	13	00:35
5,6,7	11:30	13:00	14	00:35
6,7,8	11:35	13:05	15	00:35
7,8	11:40	13:10	16	00:35
8,9	11:45	13:15	17	00:35
9,10	11:50	13:20	18	00:35
10	11:55	13:23	19	00:37
11	12:00	13:28	20	00:39
12	12:07	13:35	21	00:39
13	12:14	13:42	22	00:39
14	12:21	13:49	1	00:39
15	12:28	13:56	2	00:39
16,17	12:35	14:03	3	00:39
17,18	12:42	14:10	4	00:39
18,1	12:49	14:17	5	00:39
1,2,3	12:55	14:25	6	00:38
2,3,4	13:00	14:30	7	00:40
3,4,5	13:05	14:35	8	00:42
4,5,6	13:10	14:40	9	00:44
5,6,7	13:15	14:45	10	00:46
6,7,8	13:20	14:50	11	00:48
7,8,9	13:25	14:55	12	00:50
8,9,10	13:30	15:00	13	00:52
9,10,11	13:35	15:05	14	00:54
10,11	13:40	15:10	15	00:56
11,12	13:45	15:15	16	00:58
12,13	13:50	15:20	17	01:00
13	13:55	15:25	18	01:02
14	14:00	15:28	19	01:06
15	14:07	15:35	20	01:06
16	14:14	15:42	21	01:06
17	14:21	15:49	22	01:06
18	14:28	15:56	1	01:04
1,2	14:35	16:03	2	01:02

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
2,3	14:42	16:10	3	01:00
3,4	14:49	16:17	4	00:58
4,5,6	14:56	16:24	5	00:56
5,6,7	15:03	16:31	6	00:54
6,7,8,9	15:10	16:38	7	00:52
7,8,9,10	15:17	16:45	8	00:50
8,9,10,11	15:24	16:52	9	00:48
9,10,11,12,13	15:31	16:59	10	00:46
10,11,12,13,14	15:38	17:06	11	00:44
11,12,13,14,15	15:45	17:13	12	00:42
12,13,14,15,16	15:52	17:20	13	00:40
13,14,15,16,17	15:59	17:27	14	00:38
14,15,16,17,18	16:06	17:34	15	00:36
15,16,17,18,1	16:13	17:41	16	00:34
16,17,18,1,2	16:20	17:48	17	00:32
17,18,1,2,3	16:27	17:55	18	00:30
18,1,2,3,4	16:34	18:02	19	00:28
1,2,3,4,5	16:41	18:09	20	00:26
2,3,4,5,6	16:48	18:16	21	00:24
3,4,5,6,7	16:55	18:23	22	00:24
4,5,6,7,8	17:00	18:30	1	00:24
5,6,7,8	17:05	18:35	2	00:26
6,7,8,9	17:10	18:40	3	00:28
7,8,9,10	17:15	18:45	4	00:30
8,9,10,11	17:20	18:50	5	00:32
9,10,11	17:25	18:55	6	00:34
10,11,12	17:30	19:00	7	00:36
11,12,13	17:35	19:05	8	00:38
12,13	17:40	19:10	9	00:40
13,14	17:45	19:15	10	00:45
14,15	17:50	19:20	11	00:50
15,16	17:55	19:25	12	00:55
16	18:00	19:30	13	01:00
17	18:05	19:35	14	01:15
18	18:10	19:40	15	01:30
	18:15	19:45	16	
	18:20	19:50	17	
	18:25	19:55	18	
	18:30	20:00	19	
	18:35	20:05	20	
	18:40	20:10	21	
	18:47	20:15	22	
	18:54	20:22	1	
9	19:01	20:29	2	
10	19:08	20:36	3	
11,12	19:15	20:43	4	
12,13	19:22	20:50	5	

VEHICULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
13,14	19:29	20:57	6	
14,15,16	19:36	21:04	7	
15,16,17	19:43	21:11	8	
16,17,18,1	19:50	21:18	9	
17,18,1,2,3	20:00	21:26	10	
18,1,2,3	20:10	21:36	11	
1,2,3	20:20	21:46	12	
2,3	20:30	21:56	13	
3	20:50	22:16	14	
	21:10	22:36	15	

Anexo 6. Programación manual con 22 vehículos ordenada por rutinas.

Hora Despacho	Hora Disponible	Rutina	Espera
05:40	07:06	1	00:34
07:40	09:08	1	01:06
10:14	11:42	1	00:39
12:21	13:49	1	00:39
14:28	15:56	1	01:04
17:00	18:30	1	00:24
18:54	20:22	1	
		7	04:26
05:48	07:14	2	00:33
07:47	09:15	2	01:06
10:21	11:49	2	00:39
12:28	13:56	2	00:39
14:35	16:03	2	01:02
17:05	18:35	2	00:26
19:01	20:29	2	
		7	04:25
05:56	07:22	3	00:32
07:54	09:22	3	01:06
10:28	11:56	3	00:39
12:35	14:03	3	00:39
14:42	16:10	3	01:00
17:10	18:40	3	00:28
19:08	20:36	3	
		7	04:24
06:04	07:30	4	00:31
08:01	09:29	4	01:06
10:35	12:03	4	00:39
12:42	14:10	4	00:39
14:49	16:17	4	00:58
17:15	18:45	4	00:30
19:15	20:43	4	
		7	04:23
06:10	07:40	5	00:28
08:08	09:36	5	01:06
10:42	12:10	5	00:39
12:49	14:17	5	00:39
14:56	16:24	5	00:56
17:20	18:50	5	00:32
19:22	20:50	5	
		7	04:20

Hora Despacho	Hora Disponible	Rutina	Espera
06:15	07:45	6	00:30
08:15	09:43	6	01:06
10:49	12:17	6	00:38
12:55	14:25	6	00:38
15:03	16:31	6	00:54
17:25	18:55	6	00:34
19:29	20:57	6	
		7	04:20
06:20	07:50	7	00:32
08:22	09:50	7	01:06
10:56	12:24	7	00:36
13:00	14:30	7	00:40
15:10	16:38	7	00:52
17:30	19:00	7	00:36
19:36	21:04	7	
		7	04:22
06:25	07:55	8	00:34
08:29	09:57	8	01:03
11:00	12:30	8	00:35
13:05	14:35	8	00:42
15:17	16:45	8	00:50
17:35	19:05	8	00:38
19:43	21:11	8	
		7	04:22
06:30	08:00	9	00:36
08:36	10:04	9	01:01
11:05	12:35	9	00:35
13:10	14:40	9	00:44
15:24	16:52	9	00:48
17:40	19:10	9	00:40
19:50	21:18	9	
		7	04:24
06:35	08:05	10	00:38
08:43	10:11	10	00:59
11:10	12:40	10	00:35
13:15	14:45	10	00:46
15:31	16:59	10	00:46
17:45	19:15	10	00:45
20:00	21:26	10	
		7	04:29

Hora Despacho	Hora Disponible	Rutina	Espera
06:40	08:10	11	00:40
08:50	10:18	11	00:57
11:15	12:45	11	00:35
13:20	14:50	11	00:48
15:38	17:06	11	00:44
17:50	19:20	11	00:50
20:10	21:36	11	
		7	04:34
06:45	08:15	12	00:42
08:57	10:25	12	00:55
11:20	12:50	12	00:35
13:25	14:55	12	00:50
15:45	17:13	12	00:42
17:55	19:25	12	00:55
20:20	21:46	12	
		7	04:39
06:50	08:20	13	00:44
09:04	10:32	13	00:53
11:25	12:55	13	00:35
13:30	15:00	13	00:52
15:52	17:20	13	00:40
18:00	19:30	13	01:00
20:30	21:56	13	
		7	04:44
06:55	08:25	14	00:46
09:11	10:39	14	00:51
11:30	13:00	14	00:35
13:35	15:05	14	00:54
15:59	17:27	14	00:38
18:05	19:35	14	01:15
20:50	22:16	14	
		7	04:59
07:00	08:30	15	00:48
09:18	10:46	15	00:49
11:35	13:05	15	00:35
13:40	15:10	15	00:56
16:06	17:34	15	00:36
18:10	19:40	15	01:30
21:10	22:36	15	
		7	05:14

Hora Despacho	Hora Disponible	Rutina	Espera
07:05	08:35	16	00:50
09:25	10:53	16	00:47
11:40	13:10	16	00:35
13:45	15:15	16	00:58
16:13	17:41	16	00:34
18:15	19:45	16	
		6	03:44
07:10	08:40	17	00:52
09:32	11:00	17	00:45
11:45	13:15	17	00:35
13:50	15:20	17	01:00
16:20	17:48	17	00:32
18:20	19:50	17	
		6	03:44
07:15	08:45	18	00:54
09:39	11:07	18	00:43
11:50	13:20	18	00:35
13:55	15:25	18	01:02
16:27	17:55	18	00:30
18:25	19:55	18	
		6	03:44
07:20	08:50	19	00:56
09:46	11:14	19	00:41
11:55	13:23	19	00:37
14:00	15:28	19	01:06
16:34	18:02	19	00:28
18:30	20:00	19	
		6	03:48
07:25	08:55	20	00:58
09:53	11:21	20	00:39
12:00	13:28	20	00:39
14:07	15:35	20	01:06
16:41	18:09	20	00:26
18:35	20:05	20	
		6	03:48
07:30	09:00	21	01:00
10:00	11:28	21	00:39
12:07	13:35	21	00:39
14:14	15:42	21	01:06
16:48	18:16	21	00:24
18:40	20:10	21	
		6	03:48

Hora Despacho	Hora Disponible	Rutina	Espera
07:35	09:05	22	01:02
10:07	11:35	22	00:39
12:14	13:42	22	00:39
14:21	15:49	22	01:06
		4	03:50

ANEXO 7. Programación con Algoritmos Genéticos.

VEHÍCULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
	05:40	07:11	0	00:00
	05:48	07:19	1	00:00
	05:56	07:27	2	00:00
	06:04	07:35	3	00:00
	06:10	07:45	4	00:00
	06:15	07:50	5	00:00
	06:20	07:55	6	00:00
	06:25	08:00	7	00:00
	06:30	08:05	8	00:00
	06:35	08:10	9	00:00
	06:40	08:15	10	00:00
	06:45	08:20	11	00:00
	06:50	08:25	12	00:00
	06:55	08:30	13	00:00
	07:00	08:35	14	00:00
	07:05	08:40	15	00:00
	07:10	08:45	16	00:00
	07:15	08:50	0	00:04
	07:20	08:55	1	00:01
	07:25	09:00	17	00:00
	07:30	09:05	2	00:03
	07:35	09:10	3	00:00
	07:40	09:13	18	00:00
	07:47	09:20	4	00:02
	07:54	09:27	5	00:04
	08:01	09:34	6	00:06
	08:08	09:41	7	00:08
8,10	08:15	09:48	9	00:05
8	08:22	09:55	10	00:07
12	08:29	10:02	8	00:24
13,14	08:36	10:09	12	00:11
13,15	08:43	10:16	14	00:08
15,16	08:50	10:23	13	00:20
16,1	08:57	10:30	15	00:17
1	09:04	10:37	16	00:19
3	09:11	10:44	1	00:16
3	09:18	10:51	18	00:05
4	09:25	10:58	3	00:15
4,5	09:32	11:05	11	00:00
5	09:39	11:12	4	00:19
	09:46	11:19	5	00:19
	09:53	11:26	2	00:00
10	10:00	11:33	0	00:00
8	10:07	11:40	10	00:12
12	10:14	11:47	8	00:12

VEHÍCULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
14	10:21	11:54	12	00:12
	10:28	12:01	14	00:12
	10:35	12:08	6	00:00
	10:42	12:15	16	00:05
1	10:49	12:22	7	00:00
18	10:56	12:29	1	00:12
3	11:00	12:35	18	00:09
	11:05	12:40	11	00:00
	11:10	12:45	3	00:12
	11:15	12:50	4	00:03
	11:20	12:55	9	00:00
	11:25	13:00	17	00:00
2	11:30	13:05	15	00:00
0	11:35	13:10	2	00:09
10	11:40	13:15	0	00:07
	11:45	13:20	10	00:05
8	11:50	13:25	5	00:00
8,12	11:55	13:30	13	00:00
12	12:00	13:33	8	00:13
	12:07	13:40	12	00:13
	12:14	13:47	6	00:06
	12:21	13:54	16	00:06
	12:28	14:01	7	00:06
	12:35	14:08	18	00:00
	12:42	14:15	11	00:02
3	12:49	14:22	14	00:00
3,4	12:55	14:30	9	00:00
3,4	13:00	14:35	17	00:00
3,4,15	13:05	14:40	1	00:00
2,4,15	13:10	14:45	3	00:25
2,15	13:15	14:50	4	00:25
15	13:20	14:55	2	00:10
	13:25	15:00	5	00:00
13	13:30	15:05	15	00:25
8	13:35	15:10	13	00:05
8	13:40	15:15	12	00:00
8	13:45	15:20	0	00:00
8	13:50	15:25	6	00:03
8	13:55	15:30	16	00:01
	14:00	15:33	8	00:27
	14:07	15:40	7	00:06
18	14:14	15:47	10	00:00
18	14:21	15:54	11	00:06
	14:28	16:01	18	00:20
17	14:35	16:08	9	00:05

VEHÍCULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
17	14:42	16:15	1	00:02
	14:49	16:22	17	00:14
2	14:56	16:29	14	00:00
2	15:03	16:36	5	00:03
2,15	15:10	16:43	13	00:00
15	15:17	16:50	2	00:22
15	15:24	16:57	0	00:04
6	15:31	17:04	15	00:26
	15:38	17:11	6	00:13
	15:45	17:18	7	00:05
	15:52	17:25	10	00:05
	15:59	17:32	11	00:05
	16:06	17:39	3	00:00
	16:13	17:46	9	00:05
	16:20	17:53	1	00:05
	16:27	18:00	17	00:05
	16:34	18:07	14	00:05
	16:41	18:14	5	00:05
	16:48	18:21	13	00:05
	16:55	18:28	2	00:05
	17:00	18:35	0	00:03
	17:05	18:40	15	00:01
	17:10	18:45	4	00:00
	17:15	18:50	6	00:04
	17:20	18:55	7	00:02
	17:25	19:00	10	00:00
	17:30	19:05	8	00:00
	17:35	19:10	11	00:03
	17:40	19:15	3	00:01
	17:45	19:20	12	00:00
	17:50	19:25	9	00:04
	17:55	19:30	1	00:02
	18:00	19:35	17	00:00
	18:05	19:40	16	00:00
	18:10	19:45	14	00:03
	18:15	19:50	5	00:01
	18:20	19:55	18	00:00
	18:25	20:00	13	00:04
	18:30	20:05	2	00:02
	18:35	20:10	0	00:00
	18:40	20:15	15	00:00
	18:47	20:20	4	00:02
	18:54	20:27	6	00:04
7	19:01	20:34	10	00:01
8	19:08	20:41	7	00:13
8,11	19:15	20:48	3	00:00
8	19:22	20:55	11	00:12

VEHÍCULOS ESPERANDO	HORA DESPACHO	HORA DISPO	RU-TINA	ESPERA
	19:29	21:02	8	00:24
	19:36	21:09	17	00:01
	19:43	21:16	16	00:03
5	19:50	21:23	14	00:05
18	20:00	21:31	5	00:10
	20:10	21:41	18	00:15
	20:20	21:51	15	00:05
	20:30	22:01	12	00:00
	20:50	22:21	1	00:00
	21:10	22:41	9	00:00

ANEXO 8. Programación con Algoritmos Genéticos ordenada por rutinas.

Hora Despacho	Hora Disponible	Rutina	Espera	Almuerzo Descanso
5:40	7:11	0	0:00	
7:15	8:50	0	0:04	
10:00	11:33	0	0:00	01:10
11:40	13:15	0	0:07	
13:45	15:20	0	0:00	0:30
15:24	16:57	0	0:04	
17:00	18:35	0	0:03	
18:35	20:10	0	0:00	
	Total	8	0:18	
5:48	7:19	1	0:00	
7:20	8:55	1	0:01	
9:11	10:44	1	0:16	
10:56	12:29	1	0:12	
13:05	14:40	1	0:00	0:36
14:42	16:15	1	0:02	
16:20	17:53	1	0:05	
17:55	19:30	1	0:02	
20:50	22:21	1	0:00	1:20
	Total	9	0:38	
5:56	7:27	2	0:00	
7:30	9:05	2	0:03	
9:53	11:26	2	0:00	0:51
11:35	13:10	2	0:09	
13:20	14:55	2	0:10	
15:17	16:50	2	0:22	
16:55	18:28	2	0:05	
18:30	20:05	2	0:02	
	Total	8	0:51	
6:04	7:35	3	0:00	
7:35	9:10	3	0:00	
9:25	10:58	3	0:15	
11:10	12:45	3	0:12	
13:10	14:45	3	0:25	
16:06	17:39	3	0:00	1:21
17:40	19:15	3	0:01	
19:15	20:48	3	0:00	
	Total	8	0:53	
6:10	7:45	4	0:00	
7:47	9:20	4	0:02	
9:39	11:12	4	0:19	
11:15	12:50	4	0:03	

Hora Despacho	Hora Disponible	Rutina	Espera	Almuerzo Descanso
13:15	14:50	4	0:25	2:20
17:10	18:45	4	0:00	
18:47	20:20	4	0:02	
	Total	7	0:51	
6:15	7:50	5	0:00	
7:54	9:27	5	0:04	
9:46	11:19	5	0:19	
11:50	13:25	5	0:00	0:31
13:25	15:00	5	0:00	
15:03	16:36	5	0:03	
16:41	18:14	5	0:05	
18:15	19:50	5	0:01	
20:00	21:31	5	0:10	
	Total	9	0:42	
6:20	7:55	6	0:00	
8:01	9:34	6	0:06	
10:35	12:08	6	0:00	1:01
12:14	13:47	6	0:06	
13:50	15:25	6	0:03	
15:38	17:11	6	0:13	
17:15	18:50	6	0:04	
18:54	20:27	6	0:04	
	Total	8	0:36	
6:25	8:00	7	0:00	
8:08	9:41	7	0:08	
10:49	12:22	7	0:00	1:08
12:28	14:01	7	0:06	
14:07	15:40	7	0:06	
15:45	17:18	7	0:05	
17:20	18:55	7	0:02	
19:08	20:41	7	0:13	
	Total	8	0:40	
6:30	8:05	8	0:00	
8:29	10:02	8	0:24	
10:14	11:47	8	0:12	
12:00	13:33	8	0:13	
14:00	15:33	8	0:27	
17:30	19:05	8	0:00	1:57
19:29	21:02	8	0:24	
	Total	7	1:40	

Hora Despacho	Hora Disponible	Rutina	Espera	Almuerzo Descanso
6:35	8:10	9	0:00	
8:15	9:48	9	0:05	
11:20	12:55	9	0:00	1:32
12:55	14:30	9	0:00	
14:35	16:08	9	0:05	
16:13	17:46	9	0:05	
17:50	19:25	9	0:04	
21:10	22:41	9	0:00	1:35
	Total	8	0:19	
6:40	8:15	10	0:00	
8:22	9:55	10	0:07	
10:07	11:40	10	0:12	
11:45	13:20	10	0:05	
14:14	15:47	10	0:00	0:54
15:52	17:25	10	0:05	
17:25	19:00	10	0:00	
19:01	20:34	10	0:01	
	Total	8	0:30	
6:45	8:20	11	0:00	
9:32	11:05	11	0:00	1:12
11:05	12:40	11	0:00	
12:42	14:15	11	0:02	
14:21	15:54	11	0:06	
15:59	17:32	11	0:05	
17:35	19:10	11	0:03	
19:22	20:55	11	0:12	
	Total	8	0:28	
6:50	8:25	12	0:00	
8:36	10:09	12	0:11	
10:21	11:54	12	0:12	
12:07	13:40	12	0:13	
13:40	15:15	12	0:00	
17:45	19:20	12	0:00	2:30
20:30	22:01	12	0:00	1:10
	Total	7	0:36	
6:55	8:30	13	0:00	
8:50	10:23	13	0:20	
11:55	13:30	13	0:00	1:32
13:35	15:10	13	0:05	
15:10	16:43	13	0:00	
16:48	18:21	13	0:05	
18:25	20:00	13	0:04	
	Total	7	0:34	

Hora Despacho	Hora Disponible	Rutina	Espera	Almuerzo Descanso
7:00	8:35	14	0:00	
8:43	10:16	14	0:08	
10:28	12:01	14	0:12	
12:49	14:22	14	0:00	0:48
14:56	16:29	14	0:00	0:34
16:34	18:07	14	0:05	
18:10	19:45	14	0:03	
19:50	21:23	14	0:05	
	Total	8	0:33	
7:05	8:40	15	0:00	
8:57	10:30	15	0:17	
11:30	13:05	15	0:00	1:00
13:30	15:05	15	0:25	
15:31	17:04	15	0:26	
17:05	18:40	15	0:01	
18:40	20:15	15	0:00	
20:20	21:51	15	0:05	
	Total	8	1:14	
7:10	8:45	16	0:00	
9:04	10:37	16	0:19	
10:42	12:15	16	0:05	
12:21	13:54	16	0:06	
13:55	15:30	16	0:01	
18:05	19:40	16	0:00	2:35
19:43	21:16	16	0:03	
	Total	7	0:34	
7:25	9:00	17	0:00	
11:25	13:00	17	0:00	2:25
13:00	14:35	17	0:00	
14:49	16:22	17	0:14	
16:27	18:00	17	0:05	
18:00	19:35	17	0:00	
19:36	21:09	17	0:01	
	Total	7	0:20	
7:40	9:13	18	0:00	
9:18	10:51	18	0:05	
11:00	12:35	18	0:09	
12:35	14:08	18	0:00	
14:28	16:01	18	0:20	
18:20	19:55	18	0:00	2:19
20:10	21:41	18	0:15	
	Total	7	0:49	

ANEXO 9. CODIGO FUENTE DEL PROGRAMA.

```
unit main1;
interface

uses
  SysUtils, Windows, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Buttons, ExtCtrls, Menus, ComCtrls,
  Db, Grids,
  DBGrids, jpeg, DBCtrls, DBTables, ActnList, Math, OleCtrls,
  VCFI,
  TeEngine, Series, TeeProcs, Chart, DbChart, Variants;

type
//      define la estructura del cromosoma
  TChromo = record
    bits: array of Integer;
    totbus: integer;
    alm: integer;
    valmuerzo: array of boolean;
    vdescanso: array of boolean;
  end;

type
  TMainForm = class(TForm)
    MainMenu: TMainMenu;
    MnuSalir: TMenuItem;
    Panel2: TPanel;
    ActionList1: TActionList;
    AcGenDes: TAction;
    Panel7: TPanel;
    Panel1: TPanel;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    DBGrid1: TDBGrid;
    DataSource3: TDataSource;
    Button3: TButton;
    DBGrid2: TDBGrid;
    ProgressBar1: TProgressBar;
    Label5: TLabel;
    Label6: TLabel;
    TabSheet2: TTabSheet;
    Edit7: TEdit;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Edit8: TEdit;
    Label12: TLabel;
    Edit5: TEdit;
    Edit6: TEdit;
    Image1: TImage;
    TabSheet3: TTabSheet;
    DataSource1: TDataSource;
    DBGrid3: TDBGrid;
    DataSource2: TDataSource;
    DataSource4: TDataSource;
    Edit2: TEdit;
    UpDown2: TUpDown;
    Label1: TLabel;
    Edit1: TEdit;
    UpDown1: TUpDown;
    Label2: TLabel;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit9: TEdit;
    Edit10: TEdit;
    Label3: TLabel;
    Label4: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label13: TLabel;
    Edit11: TEdit;
    Label14: TLabel;
    Label15: TLabel;
    Edit12: TEdit;
    Label16: TLabel;
    Edit13: TEdit;
    Label17: TLabel;
    Label18: TLabel;
    procedure FileExit(Sender: TObject);
    procedure AcGenDesExecute(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure Edit2Change(Sender: TObject);
  end;

var
  MainForm : TMainForm;
  MAX_ALLOWABLE_GENERATIONS:integer;
  CROSSOVER_RATE: double;
  MUTATION_RATE:double;
  VUELTAS_DESCA: Integer;
  MIN_ALM:integer;
  MAX_VUELTAS:integer;

  g_Population: array of TChromo;
  hdespa: array of TTime;
  trecorrido: array of TTime;
  despa: array of integer;
  hdispo: array of TTime;
  bdespa: array of integer;
  despachos: integer;
  fit1: double;
  POP_SIZE: integer;
  u4,u6: double;
  TotalFitness:double;
```

implementation

uses Datos, genetico;

{\$R *.DFM}

// Asigna los vehiculos al cromosoma

function Chromo(length: integer): TChromo;

var i:integer;

begin

 SetLength(result.bits, length);

 for i:= 0 to length-1 do

 result.bits[i]:= bdespa[i];

end;

//cambia los valores en el cromosoma cuando se unen dos vehiculos

procedure arreglocromosoma(cromosoma:integer; p: integer; q:integer;despachos:integer);

var A,j:integer;

begin

 for j:=0 to despachos-1 do

 begin

 A:=g_Population[cromosoma].bits[j];

 if A=q then g_Population[cromosoma].bits[j]:=p;

 end;

end;

procedure TMainForm.FileExit(Sender: TObject);

begin

 Close;

end;

//genera tablas

procedure TMainForm.AcGenDesExecute(Sender: TObject);

var tiempo, Inicio, Hfinal,HDispo,Recorrido: TDateTime;

 null: variant;

begin

 Screen.Cursor := crHourGlass;

 DM.Tdespacho.Close;

 DM.TDespacho.EmptyTable;

 {Genera las tablas para el despacho}

 DM.Tdespacho.Open;

 DM.TDespacho.Active:=True;

 DM.TFrecuencia.Open;

 while not DM.TFrecuencia.EOF do

 begin

 Inicio:=DM.TFrecuencia['Fr_Hini'];

 Hfinal:= DM.TFrecuencia['Fr_Hfin'];

 tiempo:=DM.TFrecuencia['Fr_Tiempo'];

 recorrido:=DM.TFrecuencia['Fr_Recorrido'];

 HDispo:=Recorrido;

 while (Inicio < Hfinal) do

 begin

DM.TDespacho.InsertRecord([Null,Inicio,Null,Null,Inicio+HDispo,Null,Null,Recorrido]);

 Inicio:=Inicio + tiempo;

 end;

 DM.TFrecuencia.Next;

 end;

DM.TFrecuencia.Close;

Screen.Cursor := crDefault;

end;

procedure TMainForm.Button3Click(Sender: TObject);

var

 Null: Variant;

 INI_CRUCE: integer;

 bus, nbus, mbus: array of integer;

 A,B,C,i, j, k,l,n,w,x: integer;

 Slice: double;

 FitnessSoFar: double;

 hora, o, p,tdesca, espera: TTime;

 w1,w2,w3,w4,w5,w6: extended;

 hdispo: array of TTime;

 nvueltas: array of Integer;

 inicio: array of TTime;

 almuerzo,halmuerzo,jornada,desca: array of TTime;

 hdesca: array of TTime;

 tespera: array of TTime;

 aptitud: array of double;

 posibles: array of integer;

 posible1, posible2: TTime;

 combinables: array of integer;

 uniones: integer;

 desvia,promvueltas, min1, min2,max, e, f: double;

 Hour, Min, Sec, MSec: word;

 despachos, maximo, buses, nbuses, mbuses, y,s,t,q: integer;

 mejorbus1,peorbus1:integer;

 mejorfit1,mejorfit2:double;

 Archivo1, Archivo2, Archivo4 : TextFile;

 maxbuses:integer;

 cromosoma: integer;

 cross,b1,b2:integer;

 numbus:integer;

 uespera: array of extended;

 u3:Extended;

 esperacromo:TTime;

 horas: integer;

 tempo:Integer;

 uvueltas:extended;

 ujornada:double;

 u2,u5:double;

 u1,c2: extended;

```

c1,c3,c4: array of extended;
maxesp,minesp: extended;
maxc1, minc1, maxbus: extended;

//variables AG
Generations, cPop, crossover: integer;
temp: array of TChromo;
bits1, bits2, bits3,bits4,t1, t2, mejor: TChromo;

begin
  MAX_ALLOWABLE_GENERATIONS:=StrToInt(edit8.text);
  ProgressBar1.Max:= MAX_ALLOWABLE_GENERATIONS+1;
  ProgressBar1.visible:=True;
  POP_SIZE:=StrToInt(edit7.Text)+2;

//tasa de cruce
  CROSSOVER_RATE:= StrToFloat(edit5.text);
//tasa de mutacion
  MUTATION_RATE:= StrToFloat(edit6.text);
//vueltas antes de descanso
  VUELTAS_DESCA:=StrToInt(edit11.Text);
//Tiempo minimo de almuerzo
  MIN_ALM:=StrToInt(edit12.Text);
//Numero maximo de vueltas
  MAX_VUELTAS:=StrToInt(edit13.Text);

//Tabla TPrograma contiene la programacion
  DM.TPrograma.Active:=false;
  DM.TPrograma.Close;
  DM.TPrograma.EmptyTable;
  DM.TPrograma.Open;
  DM.TPrograma.Active:=true;
//Tabla Tdespacho tiene los despachos del dia
  DM.Tdespacho.Active:=true;
  DM.Tdespacho.First;
  DM.Tdespacho.Last;
  despachos:=DM.Tdespacho.RecordCount;

//tabla con fitness
  DM.TFitness.Active:=false;
  DM.TFitness.Close;
  DM.TFitness.EmptyTable;
  DM.TFitness.Open;
  DM.TFitness.Active:=true;

// variable para indicar desde donde inicia cruzamiento
  INI_CRUCE:=24;
// constante para tiempo de descanso entre recorridos
  tdesca:=encodetime(0,5,0,0);

  Randomize;
  setlength(hdespa,despachos);
  setlength(bdespa,despachos);
  setlength(trecorrido,despachos);

  setlength(g_population, POP_SIZE);
  setlength(temp, POP_SIZE);
  setlength(aptitud, POP_SIZE);
  setlength(uespera, POP_SIZE);
  setlength(c1, POP_SIZE);
  setlength(c3, POP_SIZE);
  setlength(c4, POP_SIZE);

  AssignFile(Archivo1, 'Archivo1.TXT');
  AssignFile(Archivo2, 'Archivo2.TXT');
  AssignFile(Archivo4, 'Archivo4.TXT');
  rewrite(Archivo1);
  rewrite(Archivo2);
  rewrite(Archivo4);

  DM.TPoblacion.Close;
  DM.TPoblacion.EmptyTable;
  DM.TRutinas.Close;
  DM.TRutinas.EmptyTable;
  DM.TRutinas.Open;
  DM.TPoblacion.Open;
  DM.TPoblacion.Active:= true;

//genera arreglos con las horas de despachos y recorridos
  DM.Tdespacho.First;
  for i:=0 to despachos-1 do
  begin
    hdespa[i]:= DM.Tdespacho['De_hora'];
    trecorrido[i]:= DM.Tdespacho['De_Recorrido'];
    DM.Tdespacho.Next;
  end;

//-----
//genera la poblacion inicial
  for n:= 0 to POP_SIZE-1 do
  begin
    fit1:=0;
//genera lista de disponibles y escoge uno aleatoriamente
    buses:=0;
    for l:=0 to despachos-1 do
    begin
      q:=0;
      if buses > 0 then
      begin
        setlength(posibles,buses);
        for i:=0 to buses-1 do
        begin
          if hdespa[i] >= hdispo[i] then
          begin
            if (nvueltas[i]<VUELTAS_DESCA) then
            begin
              posibles[q]:=bus[i];
              q:=q+1;
            end
          end
        end
      end
    end
  end

```



```

else
  inicio[A]:=hdespa[];
end;
if (g_Population[n].valmuerzo[i]=true) then
begin
  posibles[q]:=bus[i];
  q:=q+1;
end;
end;
end;
if q=0 then
begin
  setlength(bus,buses+1);
  setlength(posibles,buses+1);
  setlength(hdispo,buses+1);
  setlength(nvueltas,buses+1);
  setlength(hinicio,buses+1);
  setlength(almuerzo,buses+1);
  setlength(tespera,buses+1);
  setlength(g_Population[n].valmuerzo,buses+1);
  setlength(g_Population[n].vdescanso,buses+1);
  g_Population[n].valmuerzo[buses]:=false;
  bus[buses]:=buses;
  hdispo[buses]:=encodetime(5,0,0,0);
  nvueltas[buses]:=0;
  Posibles[0]:=buses;
  buses:=buses+1;
  q:=1;
end;
setlength(posibles,q);

//saca aleatoriamente 1 bus para desayunar o almorzar
if q >= 9 then
begin
  C:=RandomFrom(posibles);
  hdispo[C]:=hdispo[C]+encodetime(0,30,0,0);
  q:=0;
  setlength(posibles,buses);
  for i:=0 to buses-1 do
  begin
    if hdespa[] >= hdispo[i] then
    begin
      posibles[q]:=bus[i];
      q:=q+1;
    end;
  end;
  setlength(posibles,q);
end;
A:=RandomFrom(posibles);
bdespa[]:=A;
if ((nvueltas[A] >0)) then
  espera:=hdespa[]-hdispo[A]
else
begin
  espera:=0;
  inicio[A]:=hdespa[];
end;
nvueltas[A]:=nvueltas[A]+1;
hdispo[A]:=hdespa[] + trecorrido[] + tdesca;
DecodeTime(espera, Hour, Min, Sec, MSec);
tempo:=Min+Hour*60;

// si la espera es mayor a 30 minutos se vuelve almuerzo
if tempo>MIN_ALM then
begin
  if (g_Population[n].valmuerzo[A]=false) then
  begin
    g_Population[n].alm:=g_Population[n].alm+1;
    g_Population[n].valmuerzo[A]:=true;
  end
  else
  if (g_Population[n].vdescanso[A]=false) then
  begin
    g_Population[n].vdescanso[A]:=true;
  end;
end;
end;

//asigna buses a la poblacion
g_Population[n]:= Chromo(despachos);

//union de buses
uniones:=0;
for i:=0 to buses-1 do
begin
  if bus[i]<> 99 then
  begin
    setlength(combinables,0);
    setlength(combinables,buses);
    q:=0;
    for j:=0 to buses-1 do
    begin
      if hdispo[i] < hdispo[j] then
      begin
        //genero arreglo con posibles combinaciones
        if (nvueltas[j] + nvueltas[i])<=MAX_VUELTAS then
        begin
          combinables[q]:=j;
          q:=q+1;
        end;
      end;
    end;
  end;
  if q>0 then
  begin
    setlength(combinables,q);
    A:=combinables[0];
    hdispo[i]:=hdispo[A];
    bus[A]:=99;
  end;
end;
end;

```

```

    inicio[A]:=inicio[i];
    nvueltas[A]:=0;
    arreglocromosoma(n,i,A,despachos);
    uniones:=uniones+1;
end;
end;
//rutina para arreglar el numero de buses despues de la union.
nbuses:=0;
for i:=0 to buses-1 do
begin
if bus[i]<>99 then
begin
nbuses:=nbuses+1;
setlength(nbus,nbuses);
if bus[i]=nbuses-1 then
nbus[nbuses-1]:=bus[i]
else
begin
nbus[nbuses-1]:=nbuses-1;
arreglocromosoma(n,nbuses-1,i,despachos);
end;
end;
end;
end;
g_Population[n].totbus:=nbuses;
end;

//AQUI EMPIEZA EL AG
// constante para mirar numero de generaciones
Generations:= 0;
//lazo principal del AG
while (Generations < MAX_ALLOWABLE_GENERATIONS)
do
begin
TotalFitness:= 0;
mejorfit1:=0;
mejorfit2:=0;
mejorbus1:=0;
ProgressBar1.Stepl;
maxesp:=0;
maxc1:=0;
minc1:=5000;
minesp:=5000;
maxbus:=100;
for cromosoma:=0 to POP_SIZE-1 do
begin
aptitud[cromosoma]:=0;
nbuses:=g_Population[cromosoma].totbus;
fit1:=0;
u4:=0;
u5:=0;
u6:=0;
setlength(hdispo,nbuses);
setlength(nvueltas,nbuses);
setlength(hinicio,nbuses);

```

```

    setlength(nbus,nbuses);
//inicializa variables
for i:=0 to nbuses-1 do
begin
hdispo[i]:=encodetime(0,0,0,0);
Hinicio[i]:=encodetime(0,0,0,0);
nvueltas[i]:=0;
nbus[i]:=i;
end;
for j:=0 to despachos-1 do
begin
B:=g_Population[cromosoma].bits[j];
if hdespa[j]< hdispo[B] then
begin
q:=0;
setlength(posibles,nbuses);
//SELECCIONA ALEATORIAMENTE DE LOS POSIBLES
for i:=0 to nbuses-1 do
begin
if hdespa[j] >= hdispo[i] then
begin
posibles[q]:=i;
q:=q+1;
end;
end;
setlength(posibles,q);
B:=randomfrom(posibles);
g_Population[cromosoma].bits[j]:=B;
end;
if nvueltas[B] = 0 then inicio[B]:=hdespa[j];
nvueltas[B]:=nvueltas[B]+1;
hdispo[B]:=hdespa[j] + trecorrido[j] + tdesca;
end;

mbuses:=nbuses;
g_Population[cromosoma].totbus:=mbuses;
setlength(g_Population[cromosoma].valmuerzo,mbuses);
setlength(g_Population[cromosoma].vdescanso,mbuses);
setlength(nvueltas,mbuses);
setlength(almuerzo,mbuses);
setlength(desca,mbuses);
setlength(halmuerzo,mbuses);
setlength(hdesca,mbuses);
setlength(tespera,mbuses);
setlength(hdispo,mbuses);
setlength(hinicio,mbuses);
setlength(jornada,mbuses);

//inicializa variables
g_Population[cromosoma].alm:=0;
for i:=0 to mbuses-1 do
begin
hdispo[i]:=encodetime(0,0,0,0);
Hinicio[i]:=encodetime(0,0,0,0);

```

```

nvueltas[i]:=0;
tespera[i]:=0;
halmuerzo[i]:=0;
hdesca[i]:=0;
almuerzo[i]:=0;
desca[i]:=0;
g_Population[cromosoma].vdescanso[i]:=false;
g_Population[cromosoma].valmuerzo[i]:=false;
end;

//Calculos de numero de vueltas y otros parametros necesarios
para el fitness
for j:=0 to despachos-1 do
begin
B:=g_Population[cromosoma].bits[j];

if ((nvueltas[B]>0)) then
espera:=hdespa[j]-hdispo[B]
else
begin
espera:=0;
hinicio[B]:=hdespa[j];
end;
nvueltas[B]:=nvueltas[B]+1;
DecodeTime(espera, Hour, Min, Sec, MSec);
tempo:=Min+Hour*60;

// si la espera es mayor a 30 minutos se vuelve almuerzo
if tempo>MIN_ALM then
begin
if (g_Population[cromosoma].valmuerzo[B]=false) then
begin
g_Population[cromosoma].alm:=g_Population[cromosoma].alm
+1;
g_Population[cromosoma].valmuerzo[B]:=true;
halmuerzo[B]:=hdispo[B];
almuerzo[B]:=espera;
espera:=0;
decodetime(halmuerzo[B],Hour, Min, Sec, MSec);
horas:=Hour*60+min;
if horas < 11*60 then u6:=u6+0.2;
if (horas>=11*60) and (horas<=12*60) then
u6:=u6+0.2+0.8*(horas/60-11);
if (horas>12*60) and (horas<=15*60) then u6:=u6+1;
if (horas>15*60) and (horas<=990) then u6:=u6+1-
0.9*((horas-900)/90);
if (horas>990) then u6:=u6+0.1;
end
else
if (g_Population[cromosoma].vdescanso[B]=false) then
begin
g_Population[cromosoma].vdescanso[B]:=true;
hdesca[B]:=hdispo[B];

```

```

desca[B]:=espera;
espera:=0;
end;
end;
hdispo[B]:=hdespa[j] + trecorrido[j] + tdesca;
tespera[B]:=tespera[B]+espera;

DM.TPrograma.InsertRecord([hdespa[j],B,nvueltas[B],hdispo[B]
,espera,almuerzo[B] ,cromosoma,Generations]);

if Generations=MAX_ALLOWABLE_GENERATIONS-1
then
begin
if cromosoma=POP_SIZE -1 Then
writeln(Archivo1, hdespa[j],', ',B,', ',nvueltas[B],',
',hdispo[B],', ',espera,', ',almuerzo[B]);
end;
end;

//calculo del fitness
c2:=0;
promvueltas:=0;
esperacromo:=0;
uvueltas:=0;
nbuses:=0;
for i:=0 to mbuses-1 do
begin
uvueltas:=uvueltas+nvueltas[i];
if nvueltas[i]>0 then nbuses:=nbuses+1;
end;
uvueltas:=uvueltas/nbuses;
for i:=0 to mbuses-1 do
begin
if nvueltas[i]>0 then
begin
jornada[i]:=hdispo[i]-hinicio[i]-almuerzo[i]-desca[i];

DM.TRutinas.InsertRecord([i,i,nvueltas[i],jornada[i],almuerzo[i],
halmuerzo[i],desca[i],hdesca[i],tespera[i],hinicio[i],hdispo[i],
cromosoma, generations,"]);
esperacromo:=esperacromo+tespera[i];
c2:=c2+intpower(uvueltas-nvueltas[i],2);
end;
end;
DecodeTime(esperacromo, Hour, Min, Sec, MSec);
uespera[cromosoma]:=Min+Hour*60;
if uespera[cromosoma]> maxesp then
maxesp:=uespera[cromosoma];
if uespera[cromosoma]< minesp then
minesp:=uespera[cromosoma];
c1[cromosoma]:=c2;
if c1[cromosoma] > maxc1 then
maxc1:=c1[cromosoma];
if c1[cromosoma] < minc1 then

```

```

    minc1:=c1[cromosoma];
    c3[cromosoma]:=nbuses;
    if c3[cromosoma] < maxbus then
        maxbus:=c3[cromosoma];
    c4[cromosoma]:=u6/nbuses;
end;
u1:=0;
u2:=0;
u3:=0;
for l:=0 to POP_SIZE-1 do
begin
    if maxc1=minc1 then
        u1:=1
    else
        u1:=(maxc1-c1[l])/(maxc1-minc1);
    if maxesp=minesp then
        u3:=1
    else
        u3:=(maxesp-uespera[l])/(maxesp-minesp);
    u2:=1/(1+c3[l]-19);
    u4:=c4[l];
    w1:= StrToFloat(edit3.Text);
    w2:=StrToFloat(edit4.Text);
    w3:=StrToFloat(edit9.Text);
    w4:=StrToFloat(edit10.Text);
    fit1:=1/sqrt(intpower(w1*(1-u1),2)+          intpower(w2*(1-
u2),2)+intpower(w3*(1-u3),2)+intpower(w4*(1-u4),2));
//Guarda el fitness
    writeln(Archivo2,'FITNESS u1, ',u1,',u2, ',u2,', u3,', u3,', u4,
',u4,', fitness ',fit1,' ,cromosoma,',l,' ,c1,',c1[l],' ,max,',maxc1,'
,min,',minc1,' ,uespera,',uespera[l],' ,maxesp,',maxesp,'
,minesp,',minesp);
    aptitud[l]:=fit1;
    TotalFitness:= TotalFitness + fit1;
    DM.TPoblacion.InsertRecord([l,fit1,Generations]);
end;

// Crea una nueva poblacion con el metodo de la ruleta.
incluye cruzamiento y mutacion
cPop:= 0;
setlength(t1.bits,despachos);
setlength(t2.bits,despachos);
DM.QFitness.Close;
DM.QFitness.Params[0].AsInteger := Generations;
DM.QFitness.Open;
DM.QFitness.Last;
mejorbus1:=DM.QFitness['cromo'];
mejorfit1:=DM.QFitness['fitness'];
DM.QFitness.First;
peorbus1:=DM.QFitness['cromo'];
mejor:=g_Population[mejorbus1];
writeln(Archivo4,'Fitness total, ',TotalFitness);
writeln(Archivo2,'mejor , ', mejorbus1,', peor , ',peorbus1);

```

```

while (cPop < POP_SIZE-3) do
begin
    Slice:= random*TotalFitness;
    FitnessSoFar:= 0;
    for x:=0 to POP_SIZE-1 do
begin
        FitnessSoFar:= FitnessSoFar + aptitud[x];
        if (FitnessSoFar >= Slice) then
            begin
                bits1:= g_Population[x];
                break;
            end;
        end;
    end;
    Slice:= random*TotalFitness;
    FitnessSoFar:= 0;

    for x:=0 to POP_SIZE-1 do
begin
        FitnessSoFar:= FitnessSoFar + aptitud[x];
        if (FitnessSoFar >= Slice) then
            begin
                bits2:=g_Population[x];
                break;
            end;
        end;
    end;
//agrega cruzamiento
    if (random < CROSSOVER_RATE) then
begin
// cruzamiento de 1 punto
        crossover:=          RandomRange(INI_CRUCE,
despachos);
        for i:=0 to crossover-1 do t1.bits[i]:= bits1.bits[i];
        for j:=crossover to despachos-1 do t1.bits[j]:=bits2.bits[j];
        for i:=0 to crossover-1 do t2.bits[i]:= bits2.bits[i];
        for j:=crossover to despachos-1 do t2.bits[j]:=bits1.bits[j];
        if bits1.totbus > bits2.totbus then
            begin
                t1.totbus:=bits1.totbus;
                t2.totbus:=bits1.totbus;
            end
        else
            begin
                t1.totbus:=bits2.totbus;
                t2.totbus:=bits2.totbus;
            end;
        bits3.bits:= copy(t1.bits);
        bits3.totbus:=t1.totbus;
        bits4.bits:= copy(t2.bits);
        bits4.totbus:=t2.totbus;
    end
    else
begin
        bits3.bits:= copy(bits1.bits);
        bits3.totbus:=bits1.totbus;

```

```

    bits4.bits:= copy(bits2.bits);
    bits4.totbus:=bits2.totbus;
end;
//efectua mutacion
for i:= 0 to despachos-1 do
begin
    if (random < MUTATION_RATE) then
        bits3.bits[i]:= RandomRange(0,bits3.Totbus);
    end;
    for i:= 0 to despachos-1 do
    begin
        if (random < MUTATION_RATE) then
            bits4.bits[i]:= RandomRange(0,bits4.Totbus);
        end;
    end;
//Agrega la nueva poblacion
    temp[cPop].bits:= copy(bits3.bits);
    temp[cPop].totbus := bits3.totbus;
    cPop:= cPop + 1;

    temp[cPop].bits:= copy(bits4.bits);
    temp[cPop].totbus := bits4.totbus;
    cPop:= cPop + 1;
end;
for i:= 0 to POP_SIZE-3 do g_Population[i]:= temp[i];
g_Population[POP_SIZE-2]:= mejor;
g_Population[POP_SIZE-1]:= mejor;
writeln(Archivo2,'generacion, ',Generations);
Generations:= Generations + 1;
end;
CloseFile(Archivo1);
CloseFile(Archivo2);
CloseFile(Archivo4);
ProgressBar1.Position:=0;
Datasource3.Enabled:=True;
Datasource1.Enabled:=True;
DM.TPrograma.Close;
ProgressBar1.visible:=False;
end;

```

```

procedure TMainForm.Edit1Change(Sender: TObject);
begin
    DM.QFitness.Close;
    DM.QFitness.Params[0].AsInteger := StrToInt(Edit1.Text);
    DM.QFitness.Open;
    DM.QPrograma.Close;
    DM.QPrograma.Params[0].AsInteger := StrToInt(Edit2.Text);
    DM.QPrograma.Params[1].AsInteger := StrToInt(Edit1.Text);
    DM.QPrograma.Open;
    DM.QRutinas.Close;
    DM.QRutinas.Params[0].AsInteger := StrToInt(Edit2.Text);
    DM.QRutinas.Params[1].AsInteger := StrToInt(Edit1.Text);
    DM.QRutinas.Open;
end;

```

```

procedure TMainForm.Edit2Change(Sender: TObject);
begin
    DM.QFitness.Close;
    DM.QFitness.Params[0].AsInteger := StrToInt(Edit1.Text);
    DM.QFitness.Open;
    DM.QPrograma.Close;
    DM.QPrograma.Params[0].AsInteger := StrToInt(Edit2.Text);
    DM.QPrograma.Params[1].AsInteger := StrToInt(Edit1.Text);
    DM.QPrograma.Open;
    DM.QRutinas.Close;
    DM.QRutinas.Params[0].AsInteger := StrToInt(Edit2.Text);
    DM.QRutinas.Params[1].AsInteger := StrToInt(Edit1.Text);
    DM.QRutinas.Open;
end;
end.

```

ANEXO 10. CODIGO FUENTE DE LA PROGRAMACION SEMANAL SIN PREFERENCIAS

Title Despacho model

\$Ontext

\$Offtext

Sets

```
i  buses   /1*25/
j  rutinas /1*19/
k  dias    /1*5/;
```

Variables

```
x(i,j,k)      Asigna o no
beta           maximo asignamiento
landa         minimo asignamiento
z             maximo - minimo numero de rutinas;
```

binary variables x;

integer variable landa;

integer variable beta;

Equations

```
minmaxmin  valor de la funcion objetivo
rutinadiaria(j,k)  limita el asignamiento a uno
busesdiario (i,k)  limita la rutina diaria de un bus a maximo uno
betai(i)         maximo asignamiento
landai(i)        minimo asignamiento;
```

```
minmaxmin .. z =e= beta-landa;
```

```
rutinadiaria(j,k).. sum(i,x(i,j,k)) =e= 1;
```

```
busesdiario (i,k).. sum(j,x(i,j,k)) =l= 1;
```

```
betai(i).. sum((j,k), x(i,j,k)) =l= beta;
```

```
landai(i).. sum((j,k),x(i,j,k)) =g= landa;
```

```
Model Despacho / all/ ;
```

```
Solve Despacho using mip minimizing z;
```

```
Display x.l;
```

ANEXO 11. CODIGO FUENTE DE LA PROGRAMACION SEMANAL CON PREFERENCIAS

```
itle Despacho model
$Ontext
```

```
$Offtext
```

```
Sets
```

```
  i  buses   /1*25/
  j  rutinas /1*19/
  k  dias    /1*5/;
```

```
Parameters
```

```
pr(j) indice de preferencia
/ 1 0.789
  2 0.590
  3 0.865
  4 0.802
  5 0.813
  6 0.655
  7 0.551
  8 0.872
  9 0.758
 10 0.680
 11 0.870
 12 0.754
 13 0.760
 14 0.518
 15 0.785
 16 0.396
 17 0.419
 18 0.287
 19 0.526/;
```

```
Variables
```

```
  x(i,j,k)      Asigna o no
  beta          maximo asignamiento
  landa         minimo asignamiento
  z             maximo - minimo numero de rutinas;
```

```
Positive variables
```

```
  beta
  landa;
```

```
binary variables x;
```

```
Equations
```

```
minmaxmin valor de la funcion objetivo
rutinadiaria(j,k) limita el asignamiento a uno
busesdiario (i,k) limita la rutina diaria de un bus a maximo uno
betai(i)      maximo asignamiento
landai(i)     minimo asignamiento;
```

```
minmaxmin .. z =e= beta-landa;
rutinadiaria(j,k).. sum(i,x(i,j,k)) =e= 1;
```

```
busesdiario (i,k)..  sum(j,x(i,j,k)) =l= 1;
betai(i)..  sum((j,k), pr(j)*x(i,j,k)) =l= beta;
landai(i)..  sum((j,k), pr(j)*x(i,j,k)) =g= landa;
```

```
Model Despacho / all/ ;
```

```
Solve Despacho using mip minimizing z;
  Display x.l;
```