

Estructura genérica para Problemas de Ruteo de Vehículos

Integrantes:

Patsy Jones

Andrea Zilelián

Valentina Grattarola

Tutor:

Ing. Omar Viera

Taller V - 1999

INDICE

ABSTRACT 4

INTRODUCCIÓN 4

CAPÍTULO 1 - PROBLEMA DE RUTEO DE VEHÍCULOS..... 6

1.1 - CLASIFICACIÓN DE LOS PROBLEMAS DE RUTEO. 8

1.2 - ALGUNOS PROBLEMAS DE RUTEO DE VEHÍCULOS..... 9

 1.2.1 - *El Problema del Vendedor Viajante*. 9

 1.2.2 - *El Problema del Cartero Chino*. 9

 1.2.3 - *El Problema de los M Vendedores Viajantes*. 10

 1.2.4 - *El Problema de Ruteo con Deposito Unico y Múltiples Vehículos*. 10

 1.2.5 - *Problema de Ruteo con Múltiples Depósitos y Múltiples Vehículo*..... 10

 1.2.6 - *El Problema de Ruteo con Deposito Unico, Múltiples Vehículos y Demanda Estocástica*. 10

 1.2.7 - *El Problema del Cartero Chino con capacidades*. 10

1.3 - PROBLEMAS DE SECUENCIAMIENTO DE VEHÍCULOS Y TRIPULACIÓN..... 10

 1.3.1 - *El Problema de Secuenciamiento de Vehículos con Depósito Simple (VSP)* 11

 1.3.2 - *El Problema de Secuenciamiento de Vehículos con Restricciones del Largo del Camino (VSPLPR)* 11

 1.3.3 - *El Problema de Secuenciamiento de Vehículos con Múltiples Tipos de Vehículos (VSPMVT)*..... 11

 1.3.4 - *El Problema de Secuenciamiento de Vehículos con Múltiples Depósitos (VSPMD)* 11

 1.3.5 - *Administración de Trabajadores en una Locación Fija*. 11

 1.3.6 - *Administración de Vehículos y Tripulación de Tránsito Masivo* 11

1.4 - COMBINANDO RUTEO DE VEHÍCULOS Y SECUENCIAMIENTO..... 11

 1.4.1 - *Descripción de varios problemas de Ruteo de Tiempo y Secuenciamiento*..... 12

 1.4.1.1 - *Omnibus escolar*. 12

 1.4.1.2 - *Camiones con sorras con carga completa*. 12

 1.4.1.3 - *Camiones con sorras con carga parcial*. 13

 1.4.1.4 - *Vehículos de recolección de basura y de limpieza de calles*. 13

 1.4.1.5 - *Itinerario de aviones*. 13

 1.4.1.6 - *Servicios telefónico de fletes*. 13

1.5 - REQUERIMIENTOS DE DATOS. 13

1.6 - DESARROLLO Y USO FUTURO DE SISTEMAS DE RUTEO Y ADMINISTRACIÓN..... 14

1.7 - DIRECCIÓN DE LA FUTURA INVESTIGACIÓN EN EL MODELADO Y DESARROLLO DE ALGORITMOS 14

 1.7.1 - *Estrategias en el modelado* 14

 1.7.2 - *Estrategias de solución* 15

CAPÍTULO 2 - ELEMENTOS DEL PROBLEMA DE RUTEO DE VEHÍCULOS..... 16

2.1 - DEFINICIONES DEL PROBLEMA 16

 2.1.1 - *Puntos a tener en cuenta en la definición del problema*. 16

 2.1.2 - *Clases de problemas de Ruteo*..... 16

 2.1.3 - *Características operacionales de las rutas*. 18

2.2 - RESOLUCIÓN DEL PROBLEMA DE RUTEO..... 19

 2.2.1 - *Objetivos y formulación del modelo*. 19

 2.2.2 - *Estrategias de solución*. 19

2.3 - TÉCNICAS ALGORÍTMICAS 21

2.4 - BASES DE DATOS GEOGRÁFICAS Y UBICACIÓN DE VEHÍCULOS EN RUTEOS 21

 2.4.1 - *Bases de datos geográficos* 21

 2.4.2 - *Ubicación y rastreo de vehículos* 22

 2.4.3 - *Uso de la Información Geográfica en los Problemas de Ruteo* 22

CAPÍTULO 3 - SISTEMAS DE INFORMACIÓN GEOGRÁFICA (G.I.S.)..... 23

3.1 - *¿QUÉ ES UN G.I.S.?*..... 23

3.2 - *ORIGEN Y EVOLUCIÓN DE LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA*. 24

3.3 - *APLICACIONES DE LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA*..... 26

| | |
|---|-----------|
| 3.4 - VENTAJAS DE USAR G.I.S..... | 27 |
| 3.5 - LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA HOY EN DÍA | 28 |
| CAPÍTULO 4 - ESTRUCTURA DE DATOS..... | 29 |
| 4.1 - ¿ POR QUE UNA ESTRUCTURA DE DATOS?..... | 29 |
| 4.2 - ESQUEMATIZACIÓN DE ELEMENTOS PARA LA CONSTRUCCIÓN DE LA ESTRUCTURA..... | 29 |
| 4.3 - IMPLEMENTACIÓN DE LA ESTRUCTURA. | 30 |
| 4.4 - LAS ESTRUCTURAS..... | 31 |
| 4.4.1. <i>Estructura de Puntos.</i> | 32 |
| 4.5 - IMPLEMENTACIÓN EN AVENUE. | 41 |
| CAPÍTULO 5 - RESULTADOS DEL TESTEO, TIEMPOS DE CORRIDA..... | 42 |
| 5.1 - TIEMPOS DE CORRIDA. | 43 |
| 5.1.1. - <i>Juegos de datos utilizados.</i> | 43 |
| 5.1.2. - <i>Tiempos de corrida de la carga de la estructura.</i> | 43 |
| 5.1.3. - <i>Tiempos de corrida de los distintos algoritmos de Dijkstra.</i> | 44 |
| CAPÍTULO 6 - CONCLUSIONES Y TRABAJO FUTURO. | 45 |
| BIBLIOGRAFÍA. | 46 |

ABSTRACT

Existen diferentes tipos de Problemas de Ruteos de Vehículos (V.R.P.). Esto se debe a las distintas necesidades y características reales del problema a considerar, tales como la cantidad de depósitos disponibles, uno o varios; el tipo de flota: homogénea o heterogénea; la consideración de horarios de servicio; el tipo de servicio: solo entrega, solo recolección o ambas; naturaleza de la demanda; tripulación disponible y su administración, etc. Esto lleva a que se planteen algoritmos que solucionan en forma particular cada problema sobre una estructura diseñada especialmente para el mismo.

El objetivo de este Taller V, que fue solicitado por el Area de Optimización de la Empresa Ingenieros Consultores Asociados - ICA, es el desarrollo de una Estructura de Datos genérica que permita solucionar cualquier tipo de VRP, para evitar la creación de una para cada problema en particular. Este objetivo fue alcanzado, habiéndose logrado el diseño y el desarrollo de la estructura con su interfase. La implementación se realiza sobre un Sistema de Información Geográfica, ArcView¹, y el lenguaje de programación utilizado para esta implementación es Avenue, el disponible por Arcview, el cual por sus características, no es apropiado para la construcción de estructuras.

La estructura de datos implementada en este Taller V permite resolver todos los Problema de Ruteo de Vehículos con cualquier técnica de algoritmo utilizando las funciones disponibles para operar con ella.

INTRODUCCIÓN

Los Problemas de Ruteo tienen dos componentes importantes, la estructura de datos sobre la cual se corren los algoritmos, y los algoritmos en sí. En nuestro caso la preocupación se centra en la estructura de datos para los algoritmos. Cada algoritmo en particular tiene asociada una estructura que se adecua al mismo de la mejor manera, pero cualquier cambio en dicho algoritmo hace que la estructura quede obsoleta o que haya que volver a diseñarla, por lo que es importante una buena estructura de datos que soporte una gran gama de problemas. Lo anterior se debe a que la estructura depende de los elementos considerados para el Problema de Ruteo de Vehículos, y cualquier cambio en la realidad del problema trae como consecuencia la introducción de nuevos elementos, por lo que la misma debe ser modificada. Como objetivo de Taller V, se busca tener una estructura de datos con sus correspondientes funciones de manejo, que permitan montar sobre la misma todos los componentes de cualquier Problema de Ruteo de Vehículos, resolverlo y recuperar los resultados. Los elementos que se consideran en la definición de los Problemas de Ruteo son básicos para el desarrollo de esta estructura, por lo cual se realizó un estudio detallado de los mismos. Algunos de estos elementos a considerar son: la cantidad de depósitos disponibles, la cantidad de clientes a ser visitados, la demanda es conocida con anterioridad o en tiempo real, el tamaño de la flota disponible, el tipo de flota, homogénea o heterogénea, el número de conductores, etc. Este desarrollo es novedoso dada la casi no-existencia de dichas estructuras, solo encontramos una estructura de propósito similar al nuestro durante la búsqueda bibliográfica en el producto NetEngine [4].

La estructura se realiza sobre el lenguaje Avenue que es el lenguaje disponible para la herramienta ArcView, el cual es un Sistema de Información Geográfica (G.I.S.). Esta herramienta fue utilizada por requerimiento del usuario - Ing. José Comas. Un G.I.S. es una herramienta de computación para el mapeo y análisis de objetos reales y eventos que suceden en la tierra. La tecnología G.I.S. integra las operaciones comunes de las Bases de Datos como pueden ser las consultas y los análisis estadísticos con los beneficios del análisis geográfico ofrecido por la utilización de mapas. Estas cualidades distinguen G.I.S. de cualquier otro sistema de información y lo hace valioso para un gran número de empresas públicas y privadas para explicar eventos, predecir conclusiones y planificar estrategias.

¹ ArcView es un G.I.S. de E.S.R.I.

Debido a que el lenguaje de programación Avenue no brinda una gran gama de tipos de datos para el diseño de la estructura, esta se basa en el enlace de diccionarios. Es por este motivo que generalmente los tiempos de ejecución obtenidos no son muy buenos con respecto a otros lenguajes, como por ejemplo C⁺⁺. El diseño de dicha estructura se enfocó para soportar distintos algoritmos que solucionen los diferentes tipos de Problemas de Ruteo de Vehículos.

Para testear la estructura a solicitud del usuario se implementaron los siguientes algoritmos: TSP (El Problema del Vendedor Viajero), Asignación de clientes a depósitos y luego un TSP para cada Asignación.

El contenido y organización de este informe se detalla a continuación.

En el Capítulo 1 se presenta una introducción al Problema de Ruteo de Vehículos (VRP), su clasificación y sus características generales. Se plantean además las necesidades existentes y las perspectivas futuras.

En el Capítulo 2 se estudian los rasgos típicos de los Problemas de Ruteo que se han tenido en cuenta en el diseño de la estructura genérica, adaptable a todos los casos.

En el Capítulo 3 se encuentra una breve reseña de los Sistema de Información Geográfica (GIS), ya que la estructura realizada se desarrolló sobre Arcview, un software para manejar información geográfica.

En el Capítulo 4 se detalla como fue implementada la estructura de datos y que información se almacena en la misma.

En el Capítulo 5 se plantean las estrategias utilizadas para el testeado de las funciones implementadas y se presentan los resultados de los tiempos de ejecución de los principales algoritmos.

En el Capítulo 6 se desarrollan las conclusiones y resultados obtenidos en este Taller V.

A pedido del tutor de este Taller V, Ing. Omar Viera, se imprime una copia de la Documentación Técnica y de la información encontrada en Internet.

Capítulo 1 - Problema de Ruteo de Vehículos.

El Problema de Ruteo de Vehículos puede ser definido, simplídicamente, como el problema de diseñar rutas para vehículos de reparto, con capacidad conocida, los cuales son usados para operar desde un depósito para servir a un conjunto de clientes con ubicaciones y demandas conocidas. Las rutas para los vehículos son diseñadas para minimizar algún objetivo como puede ser la distancia total recorrida.

Hay muchas variantes para este problema y muchas aplicaciones en el mundo real.

El Problema de Ruteo de Vehículos ha recibido mucha atención en los últimos años debido a la creciente importancia de determinar estrategias de distribución eficientes para reducir el costo operacional en los sistemas de distribución. Es además, una de las áreas de interés del Departamento de Investigación Operativa del I.N.C.O.

Un típico Problema de Ruteo consiste en una flota de vehículos ubicados en un depósito central que debe ser administrado para proveer algún tipo de servicio a clientes geográficamente dispersos en una región de servicios. El servicio puede involucrar, por ejemplo, la entrega de mercaderías a los clientes desde el depósito, la carga y descarga de estudiantes en el ómnibus escolar, o la búsqueda y entrega de paquetes para un servicio de mensajería.

Un ejemplo simplificado es como es muestra en la Figura 1: considerar la situación de un depósito y un conjunto de clientes que son servidos por este.

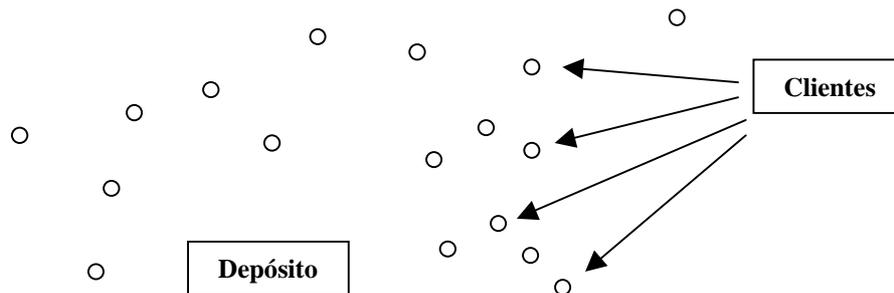


Figura 1: Un depósito con n clientes.

El planificador tiene la tarea de designar las rutas para este reparto y este problema es conocido como el Problema de Ruteo de Vehículos, lo cual esta reflejado en la Figura 2.

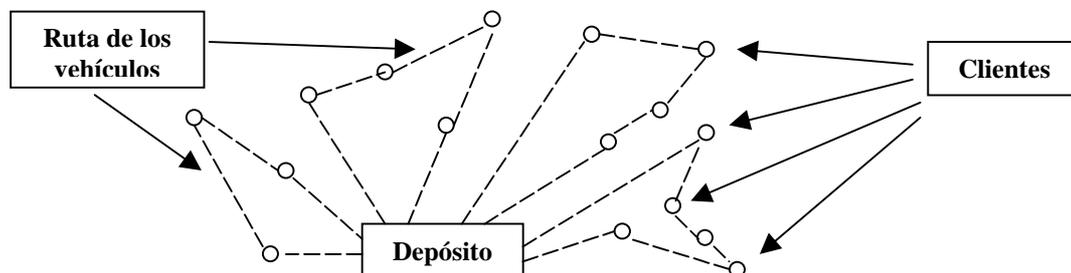


Figura 2: Resultado del V.R.P de un depósito a n clientes.

Este problema tiene una larga historia de estudios sistemáticos, los cuales comenzaron en los últimos años de la década del 50 y atrajeron mucho la atención en la literatura académica por las siguientes razones: el problema aparece en un gran número de situaciones prácticas, es teóricamente interesante y no muy fácil de resolver.

Los Problemas de Ruteo son a menudo encontrados en aplicaciones industriales con distintas restricciones dadas las diferentes características del problema tratado. Esto tiene como

consecuencia que generalmente las soluciones exactas estén fuera del alcance[1], por lo que es importante el desarrollo de algoritmos flexibles, los cuales pueden incorporar fácilmente nuevas restricciones.

Un Sistema Industrial completo cubre el proceso entero de mover materiales y requerimientos de entrada desde los proveedores a las plantas, la conversión de la entrada en productos en las plantas, el movimiento del producto a distintos almacenes o depósitos, y la eventual entrega de esos productos a los consumidores finales. Las actividades de distribución de una firma, compromete todos los movimientos y almacenamiento de materiales desde las plantas. El último paso en estos movimientos, desde los centros de distribución a los clientes, puede ser llamado transportación local o entrega, suele ser el eslabón más costoso de esta cadena de distribución [1]. Para que este paso sea llevado a cabo eficientemente, se debe unir la planificación y ejecución de las actividades de transporte de una manera racional teniendo en cuenta la economía.

El manejo efectivo de la distribución presenta una variedad de problemas de decisión en todos los niveles de estrategia, táctica y planes de operación. Decisiones relacionadas con la ubicación de los depósitos, deben ser vistas como estratégicas, mientras el problema del tamaño y la determinación de mezclas de cargas puede ser determinado tácticamente. Finalmente, sobre el nivel operacional, varias decisiones concernientes al ruteo y secuenciamiento de vehículos y a la tripulación de cada vehículo, son requeridas diariamente. Claramente la diferencia entre estrategia, táctica y planes operacionales, no deben ser interpretadas de una manera demasiado rígida, especialmente en vista de la cercana interacción entre las decisiones involucradas. Por ejemplo, la ubicación de las plantas y depósitos de flotas, tienen un gran impacto sobre las actividades de distribución de una firma. Generalmente, la ubicación de todos los depósitos es requerida como dato de entrada para la planificación de transporte de actividades locales. Por otra parte, las decisiones citadas anteriormente relacionan el costo de distribución o transporte entre varias ubicaciones geográficas. Un estudio detallado de problemas de distribución, involucra una consideración explícita de ruteo de vehículos y tamaño de la flota. Estos valores son analizados en conjunto con un plan para ruteo y secuenciamiento de vehículos asignados para la tarea de distribución.

Además de las ubicaciones de los depósitos, planes efectivos de entrega generalmente requieren entradas concernientes a una variedad de otras decisiones:

- Distribuir el tamaño del área a ser servida por cada depósito.
- Tamaño de la flota disponible de cada depósito.
- Niveles de servicios al cliente: frecuencia de entrega a cada cliente.

Dadas las decisiones anteriores, la firma debe entonces rutear y administrar sus vehículos para asignar las actividades a un costo mínimo. Este paso requiere un algoritmo de búsqueda óptimo para identificar la mejor configuración de rutas y ordenamiento de vehículos. Lo más importante de la discusión anterior, es enfatizar el impacto de varias decisiones de alto nivel en el diseño final de un Sistema de Ruteo y Secuenciamiento. Se debe remarcar que los avances recientes en los Problemas de Ruteo y Secuenciamiento sostienen el objetivo de integrar la planificación de transportes con algunas de las decisiones de alto nivel mencionadas antes.

La importancia de los problemas de distribución es evidente por la magnitud de los costos asociados a la distribución.

La salida básica de cualquier Sistema de Ruteo y Secuenciamiento es en esencia el mismo: para cada vehículo o conductor se tiene una ruta y un itinerario de tiempo, generalmente la ruta específica la secuencia de lugares a ser visitados y el itinerario identifica el tiempo en el cual las actividades para esas ubicaciones deben ser llevadas a cabo.

Los Problemas de Ruteo y Secuenciamiento se dividen en tres grupos, Ruteo, Secuenciamiento o ambos, que pueden subdividirse de acuerdo a un esquema de clasificación mas detallado.

Debido a que los Sistemas de Ruteo y Secuenciamiento comparten el mismo tipo de salida, debemos distinguirlos basándonos en otras características y suposiciones que rodean a un problema dado. Por ejemplo el problema de despachar mercadería a varias ubicaciones por una flota de vehículos desde un depósito. Si no hay restricciones de prioridad para el tiempo de entrega y si todas las mercaderías pueden ser entregadas en un periodo corto de tiempo, entonces podemos ignorar las consideraciones temporales obteniendo un Problema de Ruteo de Vehículos puro. En el caso en que el horario de visita de diferentes ubicaciones es primordial, como en el caso en que los clientes sólo aceptan la entrega dentro de un lapso de tiempo, entonces las características temporales no se deben ignorar y las restricciones de tiempo guían las actividades de ruteo y secuenciamiento. El problema

Informe Taller V

resultante se debe atacar por métodos que son diferentes de los que se usan en el primer ejemplo. En general las características y restricciones asociadas con actividades de servicios variadas llevan a distintas categorías de problemas que requieren diferente modelado.

Un Sistema de Ruteo y Secuenciamiento trata con un conjunto de entidades que requieren servicios y relaciones de precedencia entre dos entidades que establece que una deba ser atendida antes que la otra. Por ejemplo en un "Sistema de Remises" donde un pasajero debe ser recogido en una localidad y llevado a su destino. La actividad de recolección debe ser anterior a la actividad de descarga, lo cual establece una relación de precedencia entre ambas.

Un Sistema de Transito Masivo tiene un itinerario que provee tiempos definitivos para el comienzo y fin de cada viaje, controlando que el vehículo recorra el viaje completo sin interrupciones.

Si las entidades a ser atendidas no tienen restricciones de tiempo y no hay relaciones de precedencia entre ellas entonces tenemos un Problema de Ruteo. Si cada entidad tiene una ventana de tiempo, es decir, el intervalo de tiempo en el cual el servicio debe ser prestado, entonces tenemos un Problema de Secuenciamiento, de otro modo se está tratando una combinación de Ruteo y Secuenciamiento. Generalmente este último problema involucra relaciones de precedencia y ventanas de tiempo.

1.1 - Clasificación de los Problemas de Ruteo.

A continuación presentamos una tabla cuyo contenido muestra algunas de las distintas opciones existentes dependiendo de las características de los Problemas de Ruteo.

| Características | Opciones posibles |
|---|---|
| Tamaño de las flotas disponibles. | Un vehículo. Varios vehículos. |
| Tipos de las flotas disponibles | Homogénea. Heterogénea. Tipo especial de vehículo. |
| Alojamiento de vehículos | Deposito simple. Múltiples depósitos. |
| Naturaleza de las demandas | Demanda determinística. Requerimientos de demanda aleatorios. Satisfacción parcial de la demanda. |
| Ubicación de la demanda | En los nodos. Sobre los arcos. Ambos. |
| Red | No dirigida. Dirigida. Combinada. Euclidiana. |
| Restricciones de capacidad de los vehículos | Todos con la misma capacidad. Distintas capacidades. Las capacidades no son tenidas en cuenta. |
| Tiempo máximo para la ruta | Todos los tiempos iguales. Los tiempos pueden ser distintos. |
| Operaciones | Solo recolección. Solo descargas. Combinadas. |
| Costos | Costo variable. Costo fijo de operación o adquisición del vehículo. Costo común de reparto. |
| Objetivos | Minimizar el costo total. Minimizar la suma de los costos fijos y variables. Minimizar el número de vehículos requeridos. Maximizar las funciones utilizadas basándose en el servicio o la |

| | |
|--|---|
| | conveniencia. Maximizar la función utilizada basándose en las prioridades del cliente. |
|--|---|

Una consideración importante en la formulación y resolución de los Problemas de Ruteo es la carga computacional asociada con distintas técnicas de solución para estos problemas. La misma aumenta proporcionalmente con el tamaño del problema. La naturaleza de este crecimiento en los tiempos de computación como una función del tamaño de los problemas, es un tema de interés teórico y práctico. Si este crecimiento es demasiado rápido, la carga computacional se torna prohibitiva aún para problemas de tamaño moderado.

La mayoría de los Problemas de Ruteo puede ser formulado como Problema en Redes. La medida del tamaño del problema depende del número de nodos y posibles arcos de la red resultante.

Un algoritmo de límite polinomial para un problema es un procedimiento cuya carga computacional crece solo polinomialmente con el tamaño del problema en el peor caso. La clase de todos estos problemas es denotada como P. Los problemas de clase P pueden ser generalmente resueltos eficientemente.

En contraste con la clase P, hay una gran cantidad de problemas de redes y combinatoria para los cuales no se ha encontrado ningún algoritmo limitado por polinomios. Estos problemas son llamados NP-duros.

La mayoría de los Problemas de Ruteo son NP-duros [1]. Cuando nos encontramos con un problema de este tipo, frecuentemente se recurre a métodos heurísticos o procedimientos de aproximación, para obtener soluciones próximas a la óptima en lugar de buscarla. Un algoritmo heurístico es un procedimiento que usa la estructura de un problema en una forma matemática y usualmente intuitiva para proveer soluciones próximas a la óptima; un método heurístico se considera efectivo si la solución que provee es lo suficientemente cercana a esta.

1.2 - Algunos Problemas de Ruteo de Vehículos

El Problema de Ruteo básico es fácil de establecer. Esta definido por un conjunto de nodos y/o arcos a ser servidos por una flota de vehículos. No hay restricciones de cuando o en que orden estas entidades deben ser servidas. El problema es construir un conjunto factible de rutas, una para cada vehículo, a un costo mínimo. Esta ruta es una secuencia de ubicaciones que el vehículo debe visitar con una indicación del servicio a proveer.

El Ruteo de Vehículos es primordialmente un problema espacial. Se asume que no hay restricciones temporales u otras que influyan sobre la decisión de ruteo excepto aquellas que restringen el largo máximo de ruta:

Los principales Problemas de Ruteo de Vehículos son:

- El Problema del Vendedor Viajante.
- El Problema del Cartero Chino.
- El Problema de los M Vendedores Viajantes.
- El Problema de Ruteo con Deposito Unico y Múltiples Vehículos.
- El Problema de Ruteo con Múltiples Depósitos y Múltiples Vehículos.
- El Problema de Ruteo con Deposito Unico, con Múltiples Vehículos y Demanda Estocastica.
- El Problema del Cartero Chino con capacidades.

1.2.1 - El Problema del Vendedor Viajante.

Este problema requiere la determinación del ciclo de costo mínimo que pase por todos los nodos del grafo exactamente una vez.

1.2.2 - El Problema del Cartero Chino.

Este problema requiere la determinación del ciclo de costo mínimo que pase a través de cada arco por lo menos una vez. El Problema del Cartero Chino es llamado directo o indirecto según si los arcos del grafo son directos o no. Cualquiera de estas dos variantes puede ser resuelta por algoritmos limitados por polinomios.

1.2.3 - El Problema de los M Vendedores Viajantes.

Es una generalización del Problema del Vendedor Viajante donde se tienen más de un vendedor (o vehículo). Los M vehículos deben salir y retornan a un depósito común y no hay restricciones en la cantidad de nodos que visita cada vehículo excepto en que debe visitar al menos uno.

1.2.4 - El Problema de Ruteo con Depósito Único y Múltiples Vehículos.

Este problema es el clásico VRP, requiere la determinación de un conjunto de rutas de entrega para vehículos que salen y retornan a un único depósito, que atienden a todos los clientes y minimizan la distancia total recorrida. La demanda en cada nodo se asume determinística y cada vehículo tiene una capacidad dada.

1.2.5 - Problema de Ruteo con Múltiples Depósitos y Múltiples Vehículo.

Es una generalización del problema previo en que los vehículos deben servir varios depósitos en vez de uno. Todas las demás restricciones del VRP Clásico son aplicables a este tipo de problema. Además cada vehículo debe salir y retornar al mismo depósito

1.2.6 - El Problema de Ruteo con Depósito Único, Múltiples Vehículos y Demanda Estocástica.

Es idéntico al VRP Clásico excepto que las demandas no son conocidas con certeza sino que provienen de una distribución de probabilidad.

1.2.7 - El Problema del Cartero Chino con capacidades.

Este problema es el siguiente: dado una red no dirigida con arcos cuya demanda es mayor o igual a cero, se debe satisfacer la demanda de cada arco con uno de los vehículos de la flota, cada uno con una capacidad dada, encontrando los ciclos (pasando por los depósitos) que satisface todas las demandas a un costo mínimo.

1.3 - Problemas de Secuenciamiento de Vehículos y Tripulación.

Estos problemas pueden ser pensados como Problemas de Ruteo con restricciones adicionales que tienen que ver con los tiempos en que algunas actividades deben ser llevadas a cabo.

Los Problemas mencionados antes daban especial importancia a las características espaciales de las actividades desarrolladas. En particular los movimientos sucesivos de un vehículo generalmente se referían a desplazamientos en el espacio. En problemas de secuenciamiento, sin embargo, un tiempo es asociado a cada actividad. Por lo tanto los aspectos temporales del movimiento de vehículos deben ser explícitamente considerados.

En la administración de la tripulación, el principal propósito es arreglar los movimientos del personal para tenerlo disponible a los requerimientos de los vehículos. A pesar de que la Administración de Tripulación se asemeja mucho a la Administración de Vehículos, tienen otras restricciones tales como horarios de comidas de la tripulación, jornadas máximas legales, etc.

Algunos de los Problemas de Administración de Vehículos son:

- El Problema de Secuenciamiento de Vehículos con Depósito Simple (VSP).
- El Problema de Secuenciamiento de Vehículos con Restricciones del Largo del Camino (VSPLPR).
- El Problema de Secuenciamiento de Vehículos con Múltiples Tipos de Vehículos (VSPMVT).
- El Problema de Secuenciamiento de Vehículos con Múltiples Depósitos. (VSPMD)

Los Problemas de Administración de Tripulación son:

- Administración de Trabajadores en una Locación Fija.
- Administración de Vehículos y Tripulación de Tránsito Masivo.
- Administración de Tripulación Aérea.
- Problemas de registro.

1.3.1 - El Problema de Secuenciamiento de Vehículos con Depósito Simple (VSP)

Este problema requiere la repartición de nodos (tareas) de una red acíclica en un conjunto de camino de algún modo tal que cierta función de costo es minimizada. Cada camino corresponde con la administración de un único vehículo. Una función objetivo que minimiza el número de caminos, minimiza el costo de capital, ya que cada camino representa que se requiere un vehículo. Una función que minimiza el peso total de los arcos recorridos, minimiza los costos operativos, por lo que se puede usar una función objetivo combinada que minimize ambos costos.

1.3.2 - El Problema de Secuenciamiento de Vehículos con Restricciones del Largo del Camino (VSPLPR)

En el VSPLPR las restricciones son impuestas al intervalo de tiempo en que un vehículo debe pasar fuera del depósito o a la distancia que debe recorrer un vehículo antes de volver al mismo.

Estas restricciones se deben generalmente a las restricciones de combustible, mantenimiento, etc.

Mientras que el problema anterior, llamado VSP, se puede resolver usando un algoritmo limitado por polinomio este es NP-duro.

1.3.3 - El Problema de Secuenciamiento de Vehículos con Múltiples Tipos de Vehículos (VSPMVT)

El VSPMVT permite que vehículos con diferentes características estén disponibles para el servicio de las tareas. Generalmente estas características son las capacidades de los vehículos.

1.3.4 - El Problema de Secuenciamiento de Vehículos con Múltiples Depósitos (VSPMD)

En el VSPMD las tareas deben ser satisfechas desde más de un depósito. Como en el VRP cada vehículo debe partir y volver al mismo depósito y el tamaño del flete en cada depósito debe estar en un rango especificado por un mínimo y un máximo.

1.3.5 - Administración de Trabajadores en una Locación Fija.

En este problema el horario de trabajo se divide en intervalos de tiempo y se especifica la demanda de trabajadores en cada uno de estos intervalos. El Problema de Administración de Trabajadores es encontrar un conjunto de trabajadores que cubran el trabajo requerido minimizando cierto costo que podría ser la cantidad de jornadas requeridas para hacer todo el trabajo.

1.3.6 - Administración de Vehículos y Tripulación de Tránsito Masivo

El servicio de pasajeros en un sistema de tránsito masivo es definido por un cierto conjunto de líneas y administraciones de estas.

1.4 - Combinando Ruteo de Vehículos y Secuenciamiento.

La mayoría de los problemas que combinan Ruteo de Vehículo y Secuenciamiento son caracterizados por tareas de precedencia y restricciones de ventanas de tiempo. La relación de precedencia fuerza a la actividad de carga para una tarea, a preceder la actividad de descarga para la misma, las cuales deben realizarse sobre el mismo vehículo.

Las restricciones de ventanas de tiempo involucran el servicio de tareas entre las ventanas de tiempo especificadas. Una ventana de tiempo sobre una tarea de servicio requiere que la tarea sea servida durante el intervalo especificado. Cualquier ruta que involucra esta tarea particular debe asegurar que el tiempo de servicio caiga dentro de los límites de tiempo.

Algunos ejemplos de Problemas de Combinación de Ruteo de Vehículos y Secuenciamiento son:

- Omnibus escolar:
 - utilizado para una sola escuela.

- utilizado para varias.
- Camiones con sorras con cargas completas.
- Camiones con sorras con cargas parciales.
- Vehículos de recolección de basura y de limpieza de calles
 - sin restricciones
 - con restricciones de regulación de aparcamiento.
- Itinerario de aviones
- Servicio telefónico de fletes
 - sin descarga de restricciones del tiempo
 - con descarga de las restricciones del tiempo

1.4.1 - Descripción de varios problemas de Ruteo de Tiempo y Secuenciamiento.

1.4.1.1 - Omnibus escolar.

En este problema hay un número de escuelas, donde cada una tiene un conjunto de paradas de ómnibus con un número asignado de estudiantes en cada parada, además de un tiempo de comienzo y fin fijos con respecto a las ventanas de tiempo para la subida y bajada de estudiantes. La ventana de tiempo antes de la hora de comienzo de la escuela involucra el intervalo permitido para la llegada de estudiantes a la escuela en la mañana y la ventana de tiempo después de la hora de finalización del horario escolar es el intervalo de tiempo permitido asociado con la partida de los estudiantes al final del día escolar. El principal objetivo de este problema es minimizar el "costo de transporte". Para una escuela con una flota de vehículos alquilados, un objetivo conveniente para el costo del transporte es minimizar el número total de vehículos requeridos. Para una escuela que es propietaria de los vehículos, un objetivo conveniente es minimizar una combinación de los costos de operación de las flotas y el número de vehículos usados.

La mayoría de los problemas en esta área están concernidos con el *problema de carga simple*, esto es el problema en el cual se cargan y descargan estudiantes de una misma escuela. En las áreas rurales, *el problema de mezclas de cargas* puede ocurrir. En estos problemas, los estudiantes de varias escuelas son recogidos por un ómnibus, algunos estudiantes son dejados en una escuela mientras que otros son transportados a la próxima escuela, en el trayecto entre escuelas otros estudiantes pueden ser recogidos.

1.4.1.2 - Camiones con sorras con carga completa.

Un problema de la distribución comercial común involucra el ruteo y administración de tiempo de la porción de la sorra de los camiones donde se asume que las sorras están completamente cargadas. El término "carga completa" significa que una sorra esta asignada a un camión y es transportado directamente desde un punto de carga (origen) a un punto de descarga (destino). La carga sobre una sorra tiene un destino único y especificado y no debe ser volcado en ubicaciones diferentes al destino. La capacidad del camión es una sorra, debido a que cada una es transportada desde el origen a su destino, este problema involucra restricciones de precedencia.

Las demandas son especificadas en términos del número de viajes de sorras entre pares destino-origen. Una vez que se tiene el dato de la demanda, se debe minimizar el costo de la distribución total para manejar todas las demandas origen-destino.

Informe Taller V

1.4.1.3 - Camiones con sorras con carga parcial.

Este problema es similar al anterior excepto que la demanda de cada par origen-destino no necesita ser una sorra de carga completa. Como consecuencia, la carga sobre una sorra debe ser descargada entre diferentes orígenes y destinos.

Las restricciones de este problema son las relaciones de precedencia para cada par origen-destino, las ubicaciones de las ventanas de tiempo, y las reglas de trabajo de los conductores como el número máximo de horas/semanas que los conductores pueden manejar y las restricciones de las horas de descanso.

1.4.1.4 - Vehículos de recolección de basura y de limpieza de calles.

Para estos problemas, un conjunto de segmentos de calles es especificado como servicio necesario. El problema básico es cubrir cada segmento de calle por un vehículo de manera que el número de vehículos usados sea mínimo. Otro objetivo más ambicioso es minimizar el número total de tiempo de los vehículos para un tamaño de flota fijo o variable. No hay relaciones de precedencia sobre las entidades a ser servidas. Las ventanas de tiempo para los Problemas de Ruteo y Secuenciamiento de estos problemas corresponden a las regulaciones de los aparcamientos sobre las calles. En la mayoría de los casos, no hay ventanas de tiempo sobre las demandas para los problemas de recolección de basura.

1.4.1.5 - Itinerario de aviones.

Las aerolíneas coordinan el itinerario de sus aviones con el proceso de generar sus horarios. La generación de estos tiene las siguientes consideraciones: el número de pasajeros esperado, la frecuencia de servicio deseado, viajes con o sin escalas, etc.

1.4.1.6 - Servicios telefónico de fletes.

En estos problemas, los clientes llaman a un despachador o piden un servicio de respuesta. Cada cliente especifica un punto distinto de carga y descarga, y quizá, el tiempo deseado de carga y descarga. Si todos los clientes piden servicio inmediato, entonces ruteo y secuenciamiento es hecho en tiempo real y el problema es un *Problema dinámico o de tiempo real*. Si todos los clientes llaman con anticipación, entonces la demanda de clientes es conocida antes de que cualquier ruteo o secuenciamiento sea llevado a cabo, entonces este problema es conocido como el *Problema del suscriptor o estático*. Ambos problemas tienen relaciones de precedencia porque el pedido de un cliente debe ser cargado antes de ser entregado.

1.5 - Requerimientos de datos.

Muchos Problemas de Ruteo y Secuenciamiento tienen características que los agrupan en las siguientes categorías y como consecuencia de esto, generan requerimientos en las mismas. Estas categorías son: distancias y/o tiempo de viaje, costos y requisitos de servicio. Sin embargo, en muchos casos, un modelo es elegido considerando principalmente la minimización de los requerimientos. Por ejemplo, el clásico Problema de Ruteo de Vehículos tiene, como objetivo, minimizar la distancia recorrida por una flota fija de vehículos con capacidad limitada sujeta a satisfacer demandas. Los requerimientos de datos son distancias, demandas, capacidad de los vehículos y el tamaño de la flota. Todas estas cantidades son relativamente fáciles de obtener. Sin embargo, el uso de distancias en la función objetivo y las restricciones del tamaño de la flota, en muchos casos, sirve como un remplazo de una compleja función de costo que incluye costos fijos que varían con el número de vehículos, con las distancias y el tiempo en que los vehículos están en servicio. Las restricciones, además, pueden ser una representación simplificada de las que ocurren en la práctica.

Los requerimientos de una eficiente técnica de solución pueden limitar la aplicabilidad de un modelo a los problemas del mundo real, extensos requerimientos de datos puede además impedir que ciertos modelos sean utilizados.

La primer categoría de datos, la distancia, es probablemente la más fácil de manejar. Es en muchos casos un remplazo del tiempo de viaje. El tiempo de viaje requerido entre un par de

ubicaciones es una variable aleatoria que depende por un lado de la hora del día, y por otro lado de la ubicación del viaje.

En muchos casos, tal como los Problemas de Secuenciamiento de Vehículos y Tripulación que surgen de tablas de tiempo y en algunos Problemas de Ruteo de Vehículos, los requerimientos de demandas y servicios están bien definidos y son fáciles de obtener. Sin embargo, en un gran número de aplicaciones, por ejemplo en la recolección de residuos y entrega de combustible, la demanda debe ser estimada.

1.6 - Desarrollo y uso futuro de Sistemas de Ruteo y Administración

En la actualidad, las organizaciones están comenzando a automatizar sus Actividades de Ruteo y Administración. Como las ventajas de estos sistemas son cada vez más conocidas, las investigaciones en la realización de nuevos sistemas se incrementan significativamente.

El decrecimiento en los costos de la computación en general, hacen que la automatización de los sistemas sea más económica.

Desafortunadamente, una de las mayores debilidades en la utilización de modelos de Ruteo y Secuenciamiento es la carencia de software efectivo de propósito general para estos problemas.

La mayoría de los sistemas de propósito general utilizados hasta la fecha han tenido por lo general dos grandes faltas: la incapacidad para manejar problemas particulares y condiciones adicionales, y la falta de adecuación de los sistemas de bases de datos especialmente en las interfaces con el usuario. Para solucionar esto, en el futuro, los sistemas de software deben incorporar los siguientes ingredientes vitales: una flexible e interactiva interface con el usuario y un sistema de base de datos que sea fácil de entender y manipular.

La evolución de los Sistemas de Ruteo y Secuenciamiento está muy relacionada con el desarrollo futuro de algoritmos.

1.7 - Dirección de la futura investigación en el modelado y desarrollo de algoritmos

Organizaciones comprometidas en actividades de distribución están constantemente encontrando nuevos problemas que difieren en su conjunto de condiciones, estructuras de costos y criterios de performance. Estos problemas necesitarán el desarrollo de nuevas formulaciones matemáticas, metodologías de solución, e implementaciones. Por otro lado, nuevos desarrollos en programación matemática constantemente sugiere y permite refinar técnicas de solución por cada problema ya estudiado, por ejemplo el problema del viajante.

En 1973, unos pocos métodos generales estaban disponibles para el Problema de Ruteo de Vehículos estándar (VRP). Siete años después, se tenían por lo menos cinco métodos más [1]. Además, estos métodos representan significativos avances en la resolución del problema exacta y eficientemente.

1.7.1 - Estrategias en el modelado

Todos los Problemas de Ruteo y Secuenciamiento tienen esencialmente la misma salida, la diferencia entre estos problemas está en las suposiciones realizadas y la naturaleza de los requerimientos de entrada. La lógica entre las suposiciones realizadas sobre modelo y su rol en convertir un problema complejo del mundo real en uno estructurado, son tan importantes como las estrategias algorítmicas adoptadas para resolver el mismo.

Algunas de las preguntas que pueden ser planteadas sobre el modelado son las siguientes:

- Dado un particular sistema de distribución del mundo real, qué importancia tienen las consideraciones de tiempo?
- En particular, cuándo podemos ignorar todas las condiciones de tiempo para que el modelo clásico de Ruteo de Vehículos pueda ser usado?

- Cuándo pueden ser manejadas las condiciones de tiempo por una simple condición en el total del tiempo de ruteo?
- Cuándo las condiciones de tiempo deben ser explícitamente consideradas para que un modelo de ruteo y secuenciamiento deba ser usado?

1.7.2 - Estrategias de solución

a) Algoritmos exactos. Excepto el problema del viajante, el esfuerzo en desarrollar algoritmos exactos para Problemas de Ruteo y Secuenciamiento NP-duros ha sido muy limitado [1]. Hasta para los Problema de Ruteo de Vehículos la existencia de algoritmos exactos es todavía inferior que para el problema del viajante. Con esto podemos concluir que el área de técnicas de solución exacta para el ruteo y secuenciamiento está todavía en la etapa de la niñez.

b) Algoritmos heurísticos. Dada la complejidad de los Problemas de Ruteo y Secuenciamiento y la gran cantidad de problemas encontrados en la práctica, los métodos heurísticos continúan siendo los algoritmos dominantes en la investigación de tales problemas.

Capítulo 2 - Elementos del Problema de Ruteo de Vehículos.

El área de Ruteo de Vehículos se distingue por una amplia interacción entre las técnicas de algoritmos y el desarrollo de Sistemas de Ruteo efectivos para la industria.

La clave del éxito de los Problemas de Ruteo de Vehículos está en el poder de los algoritmos y la capacidad de las computadoras de resolver el problema preservando el realismo del modelo. En los últimos años se ha acumulado una gran experiencia práctica, cuyo principal logro ha sido la captura de suficientes características del ambiente real de distribución para permitir a los procesos obtener soluciones útiles.

La mayoría de las aplicaciones exitosas han sido por la combinación de un modelado cuidadoso, heurísticas inteligentes y conocimiento del problema a ser estudiado; es por esta razón que es muy importante determinar las características de los Problemas de Ruteo estudiando los elementos que lo definen. El estudio de los mismos es desarrollado en este capítulo.

2.1 - Definiciones del problema

2.1.1 - Puntos a tener en cuenta en la definición del problema.

El primer paso en la aplicación de técnicas de ruteo, es claramente el proceso de definir el problema. Al definirlo se debe considerar la naturaleza del problema y el tamaño del ambiente de la empresa en cuestión.

Alguno ejemplos de la información importante a tener en cuenta en las actividades de distribución para la definición son:

- Tamaño de la flota disponible.
- Número de conductores.
- Número de rutas recorridas diariamente y promedio de embotellamientos.
- Actividades entre o dentro de ciudades.
- Costo anual total de las actividades de distribución.
- Costos de vehículos y tripulación como porcentaje del costo total.
- Futuras demandas y áreas de servicio.
- Capacidad y soporte de las computadoras para el ruteo.
- Integración de ruteo con otras actividades.

2.1.2 - Clases de problemas de Ruteo

Entre otras cosas un modelo requiere una clasificación de los Problemas de Ruteo en relación a la naturaleza de las actividades de distribución involucradas. Las actividades básicas de los clientes de la mayoría de las distribuciones están dentro de las siguientes categorías:

- Solo entrega o solo recolección.
- Entrega y recolección combinadas.

Dentro de los problemas de entrega y recolección se pueden distinguir los siguientes casos:

- Muchos lugares de entregas para pocos lugares de recolección.
- Pocos lugares de entregas para mucho lugares de recolección.

Otra categoría de problemas es aquella donde las actividades ocurren en los arcos y no en los nodos.

Los Problemas de Ruteo basados en la importancia relativa de los aspectos temporales y espaciales de un problema, se dividen en ruteo, secuenciamiento y ambos. Sin embargo, todos estos problemas resultan en un conjunto de administración de vehículos, por lo que, la diferencia no se puede hacer basada en la salida final del sistema. Una dificultad al encontrar un esquema de clasificación adecuada es cuando las restricciones del problema o la técnica de solución debe ser

Informe Taller V

usada como base para la clasificación. Por ejemplo, se podría empezar con un Problema de Ruteo con ventanas de tiempo. Para cierta distribución de las ventanas puede ser factible resolver el problema con un algoritmo que busque una buena configuración espacial de las rutas. En este caso el uso de un algoritmo de ruteo puro, refleja una decisión de reclasificar el problema basado en la flexibilidad de las restricciones de tiempo.

En vez de hacer una clasificación en términos generales, se puede clasificar el problema según diversos puntos. Estos puntos especifican la naturaleza de la demanda, la flota de vehículo, la disponibilidad de la tripulación, reglas de trabajo y las estructuras de pago, y se listan a continuación.

| | |
|--|---|
| Según la naturaleza de la demanda: | Solo entrega o recolección. Entrega y recolección combinadas. ¿Se deben servir todas las demandas?. Prioridad para los clientes |
| Según la información de la demanda: | ¿Toda la demanda es conocida con anterioridad? ¿Frecuencias fijas para las visitas? Flujo de demanda en tiempo real. |
| Según la flota de vehículos: | Flota homogénea o tipos múltiples de vehículos. Vehículos con restricciones de peso y capacidad. Vehículos con compartimentos. Compatibilidad entre la carga y el tipo de flota. Dependencias entre el tipo de vehículos y los sitios de entrega. Tamaño fijo o variable de la flota. Flotas basadas en depósitos simples o terminales múltiples. |
| Según la disponibilidad de la tripulación: | Estructura de pago. Duración del jornal. Mínimo y máximo de horas trabajadas. Opción de horas extras. Número fijo o variable de conductores. Lugar y hora de salida de los conductores. Almuerzo y otros descansos. ¿Son permitidos los viajes de varios días? |
| Según los requerimientos de Secuenciamiento: | Asignación de clientes a un día de la semana. Ventanas de tiempo para entrega y recolección. Tiempos para la carga y descarga. |
| Según los requerimientos de datos: | Información geográfica y de rutas. Direcciones y ubicaciones de los clientes. Tiempos de viaje. Información de la ubicación de los vehículos. |

Para ilustrar las complejidades existentes en los problemas de ruteo, examinamos las categorías según las características de las demandas con respecto a los siguientes puntos:

- Según la selección de clientes a ser atendidos.
En el VRP clásico todos los clientes deben ser atendidos, sin embargo, en otros problemas los clientes pueden dividirse en diferentes clases: los que deben ser atendidos en ese período, los que pueden ser atendidos y aquellos que deberían ser retrasados hasta un período futuro (ruteo periódico).
- Según la incertidumbre en el tamaño de las entregas a los clientes.
Un ejemplo de este punto es el Distribuidor de bebidas, la demanda exacta de los clientes visitados en una ruta dada no se conocen previamente, sino que el conductor vende tanto producto como puede en cada lugar. Este problema se conoce como Problema de Ruteo de Vehículos Estocástico.
- Según la opción de satisfacer parcialmente la demanda.

Por ejemplo la Distribución de los gases industriales cuando el nivel de tanques de los clientes es conocido a través de la lectura del metraje, pero el sistema de distribución debe decidir si satisfacer las necesidades del cliente solo parcialmente para la economía del ruteo. En este caso el tamaño de la entrega es una variable de decisión del modelo. Algunas ubicaciones pueden ser elegidas como sitios de volcado, donde el vehículo descarga los restos de gas antes de volver al depósito.

- Según el tiempo en el cual la información de la demanda está disponible. En el VRP clásico todas las demandas son conocidas con anterioridad, sin embargo, en el ruteo dinámico las demandas son conocidas en tiempo real, a medida que la actividad de ruteo se desarrolla. En un escenario común de distribución la mayoría de la demanda para un día dado se conoce con anterioridad, mientras que en los servicios de mensajería la demanda es especificada dinámicamente a medida que avanza el día.

2.1.3 - Características operacionales de las rutas.

Una vez que las características del problema son definidas se debe investigar las características operacionales de las actividades de ruteo, por lo cual es importante especificar las restricciones que manejan las configuraciones de las rutas. Las rutas recorridas en cierto día están sujetas a un número de restricciones impuestas por los vehículos, la agenda del conductor, las preferencias de los clientes o la región geográfica de reparto.

Los distintos grupos de restricciones que existen sobre la configuración de las rutas son los siguientes:

El primer grupo se refiere a las restricciones temporales de las rutas como:

- Duración de la ruta.
- Ubicaciones iniciales y finales de las rutas del conductor.
- Tiempo inicial de las rutas.
- Viajes en la noche y leyes laborales asociadas.
- Descansos intermedios.

Normalmente, el punto de comienzo y fin de una ruta es el depósito, sin embargo, una elección debe ser realizada en problemas con múltiples depósitos o cuando ciertos vehículos comienzan o terminan sus viajes en la casa del conductor. En estos problemas, la elección de la base para un conjunto de rutas debe de ser una variable de decisión que es optimizada al mismo tiempo que son diseñadas las rutas. Además para administrar las rutas de los conductores se debe determinar los tiempos de comienzo y duración considerando las ventanas de tiempo de los clientes y los descansos de los conductores. En algunos problemas, la determinación de los tiempos de comienzo requiere especial atención. Por ejemplo, estos tiempos deben ser escalonados debido al límite en el número de vehículos que pueden ser cargados simultáneamente, este caso se da en el problema del reparto de diarios donde los tiempos de comienzo están relacionados con la tasa de producción de la imprenta, la cual pone restricciones en la carga de los camiones.

El segundo grupo de restricciones que se deben tener en cuenta tienen que ver con la carga del vehículo usado.

- Restricciones de peso y volumen sobre la carga de los vehículos.
- Restricciones de carga.

El VRP clásico permite restricciones en el total de la carga permitida por el vehículo, en la práctica esta simple capacidad debe ser remplazada por restricciones de peso y volumen.

Debido a que la configuración de las rutas puede estar fuertemente ligada a cierta geografía o territorios de manejos definidos anticipadamente, el tercer grupo se refiere a este tipo de características.

- Territorios asignados o regiones de entrega.
- Límites de las regiones naturales o legales.
- Agrupamiento de clientes.
- Reglas para dividir cargas.

Un ejemplo simple de esto es cuando el conductor esta restringido a un territorio geográfico fijo. Esto podría ser porque el cliente espera ver el mismo conductor de una visita a la próxima estableciendo territorios de clientes a los conductores.

El cuarto grupo se refiere a las restricciones de uso y modificación de las rutas como:

- Importancia de las rutas balanceadas.
- Rutas múltiples por vehículo.
- Uso de rutas fijas y paradas fijas sobre ellas.
- Modificaciones de las rutas dependiendo de los nuevos pedidos.

El último grupo se relaciona con las restricciones referidas a las configuraciones de las rutas.

- Reglas especiales para ciertos trechos de la ruta.
- Calle en un solo sentido.
- Evitar los giros en U, u otras reglas de seguridad.

Por ejemplo, en problemas de ruteo de arcos, cuando un vehículo debe atravesar calles de una red de rutas se debe de tener especial atención al camino que el vehículo sigue en la red para descartar por ejemplo vueltas en U y otras prácticas de manejo inseguras.

2.2 - Resolución del problema de ruteo.

2.2.1 - Objetivos y formulación del modelo.

Dos puntos claves para construir un modelo apropiado para el problema son el nivel de detalle incorporado al modelo y la naturaleza de la función objetivo.

El nivel de detalle incorporado al modelo involucra la elección entre la exactitud y la tratabilidad. El modelo debe reflejar la economía del proceso de distribución, lo cual debe ser detectado tempranamente. Un ejemplo simple puede ser determinar si el objetivo real es minimizar los costos relativos a la distancia sobre la flota de vehículos o simplemente sobre la cantidad de vehículos usados. En aplicaciones donde los conductores tienen un mínimo de horas de trabajo por día garantizadas, la economía real se asocia con la eliminación de conductores. Ahorros en el número de conductores (o vehículos) pueden ser acompañados por el uso de una cierta cantidad de horas extras lo cual introduce la necesidad de imponer al modelo restricciones en el largo de la jornada.

En varios problemas de Ruteo la función objetivo tiene naturaleza multi-atributo. En general como las restricciones del problema se van complicando, y además las técnicas de solución están compuestas de submodelos interactivos, la función objetivo comienza a asumir el rol de una apropiada elección de control sobre la calidad de la solución, mas que la representación exacta del costo total de operación.

2.2.2 - Estrategias de solución.

Los sistemas comerciales de ruteo frecuentemente incluyen un amplio rango de rasgos, la mayoría se usa para la distribución comercial de mercaderías. Muchos de los paquetes resuelven una versión del Problema de Ruteo de Vehículos Clásico incluyendo solo entrega o solo recolección con una variedad de restricciones, excepcionalmente algunos manejan problemas de recolección y entrega con ventanas de tiempo.[2]

La siguiente tabla provee algunos rasgos disponibles en paquetes comerciales:

- Rasgos
 - Diferentes tipos de vehículos.
 - Entrega y recolección.
 - Ventanas de tiempo.
 - Tiempos de carga y descarga.
 - Velocidades variables.

- Objetivos
 - Minimizar la distancia.
 - Minimizar el tiempo de viaje.
 - Minimizar la flota.
 - Minimizar el costo total.
- Información de la ruta
 - Capacidad y duración límite.
 - Uso del tiempo extra.
 - Tiempo de comienzo y finalización.
 - Equipos de dos conductores.
 - Paradas fijas en las rutas.
 - Vehículos multirutas.
- Salida
 - Agenda e itinerario de la ruta.
 - Reporte de utilización del vehículo.
 - Reporte de utilización de los conductores.
 - Gráficos para las rutas.
 - Redes de rutas cubiertas.

Esta tabla reúne los rasgos de los paquetes disponibles comercialmente, pero estos no existen simultáneamente en uno solo, sino que son la unión de los rasgos de diferentes paquetes.

Si un problema tiene características no estándares, hay que decidir si un sistema hecho totalmente a medida es justificado, o si el problema puede ser apropiadamente modificado o descompuesto para sacar provecho de los paquetes estándares disponibles en el mercado.

Como ejemplo consideramos las operaciones de un servicio de mensajería metropolitana, que ofrece en el día la recolección y entrega de paquetes entre parejas de origen-destinatario. El área de entrega se divide en un número de regiones o territorios, cada uno asignado a un conductor. La jornada de trabajo se divide en cuatro turnos. Durante cada uno de estos un vehículo debe cubrir su territorio con una ruta basada en el depósito llevando a cabo actividades de recolección y entrega como sea necesario. Algunos paquetes son recolectados y entregados en el mismo turno, mientras que otros son solo recolectados en un intervalo, llevados al depósito y entregados en el siguiente turno.

A primera vista esto parece ser un complicado Problema de Ruteo de Vehículos. Sin embargo las siguientes observaciones pueden ayudarnos a simplificar el problema considerado.

- La capacidad del vehículo no está limitada. Se asume que los paquetes son lo suficientemente pequeños para la capacidad del vehículo.
- Las duraciones de las rutas están limitadas por el largo del turno en el cual ocurren.
- La demanda es conocida con al menos un turno de anterioridad por lo tanto cambios en tiempo real de las rutas no son frecuentes.
- Los territorios son estables permitiendo escasas superposiciones entre las regiones operativas de los diferentes conductores.

Dadas estas características uno puede descomponer el problema por turnos en Problemas de Ruteo separados. El Problema de Ruteo en cada intervalo no tiene restricciones de capacidad, pero sí límites estrictos en la duración de la ruta. Sin embargo, se diferencia del VRP clásico en que tanto la entrega como la recolección ocurren en la misma ruta. Ya que la capacidad del vehículo no es un punto a considerar, la principal restricción es la relación de precedencia entre las tareas, la recolección debe preceder a su entrega asociada. El despachante debe decidir que paquetes son recolectados y entregados en la misma ruta basados en las necesidades del cliente y las restricciones de duración de la misma. Si el número de paquetes no es muy grande, el VRP clásico puede ser un muy buen comienzo para la construcción de las rutas asociadas a cada turno.

Para concluir se debe decidir cuando un algoritmo de propósito general puede ser usado adecuadamente o adaptado para dar soluciones aceptables posiblemente con la ayuda de otros métodos heurísticos o procedimientos adecuados luego del proceso. Si no se logra esto, un algoritmo debe ser diseñado específicamente para este problema.

2.3 - Técnicas algorítmicas.

En los últimos años el desarrollo de algoritmos de Ruteo de Vehículos ha tenido la habilidad de enriquecer los modelos previos para manejar ambientes de operación mas realistas y adaptables. De hecho se puede argumentar que el mayor esfuerzo para el desarrollo de algoritmos se originó en aplicaciones prácticas.

Se pueden identificar tres tendencias para ganar realismo en modelos de ruteo:

- Elegir un buen método heurístico de Ruteo de Vehículos y luego enriquecerlo para manejar algunas de las restricciones reales sobre la configuración del ruteo.
- Usar técnicas de programación matemática para construir marcos para la solución de problemas de Ruteo de Vehículos complicados. Dos clases de modelos que probaron ser exitosos en aplicaciones fueron aquellos basados en las formulaciones del Método de Asignación Generalizada y formulaciones de Particiones de Conjuntos.
- Una tercer área de investigación se ha enfocado en el diseño de lo que puede ser llamado "heurísticos secuenciales" para Problemas de Ruteo, en el que el problema debe ser descompuesto en una secuencia de subproblemas que deben ser resueltos secuencialmente con diferentes algoritmos. Estos desarrollos son maneras convenientes de atacar ruteos complejos.

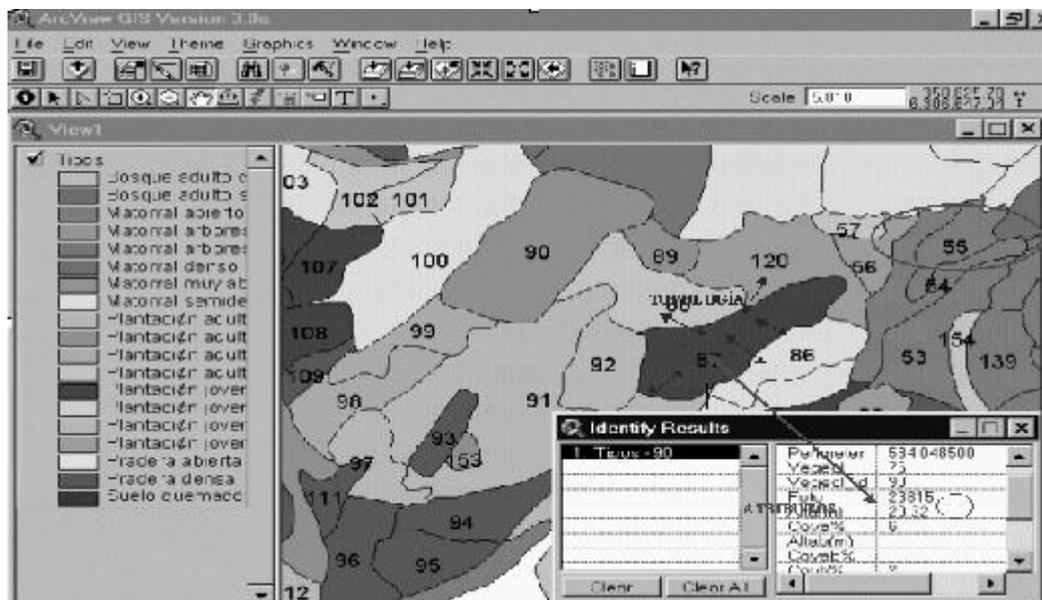
2.4 - Bases de datos geográficas y ubicación de vehículos en ruteos

El uso de información geográfica exacta sobre la ubicación de los clientes y la red de calles han sido el principal estudio en la distribución. En los últimos años, el principal desarrollo ha ocurrido en las dos áreas: la de bases de datos geográficas y ubicación de vehículos y rastreo, los cuales se espera que impacten significativamente en los Sistemas de Ruteo.

2.4.1 - Bases de datos geográficos

En los últimos años se han realizados numerosos esfuerzos para que la información geográfica este electrónicamente disponible, esto fue necesario debido al aumento del uso de mapas en las diferentes áreas, al avance de la tecnología y a la necesidad de integrar los datos geográficos con la información operativa de las empresas. Por este motivo es que existen en el mercado Sistemas de Información Geográficos, los cuales son cada vez más usados por las ventajas que estos brindan.

Una vez que una base de datos detallada se ve como una base general en vez de solo un mapa, es fácil tener una visión amplia de las facilidades que ésta puede ofrecer.



2.4.2 - Ubicación y rastreo de vehículos

En ausencia de otro medio de comunicación, los conductores deben llamar a la base o depósito frecuentemente para informar su ubicación actual, por lo que la posición del vehículo puede ser solamente estimada en el intervalo entre dos llamadas.

Existen otras técnicas más avanzadas de ubicación de vehículos como:

- Sensores de navegación pueden combinar información sobre la velocidad del vehículo y cambios en la dirección con los datos disponibles sobre la ruta recorrida.
- Pulsos de radio de baja frecuencia son enviados desde pares de estaciones al vehículo lo cual lo relaciona con la estación base. Esto permite a la computadora ubicar el vehículo sobre un camino hiperbólico basado sobre los tiempos de las dos señales recibidas por el vehículo.
- Una señal disparada desde la base al vehículo manda una señal a un par de satélites, los cuales luego lo devuelven a la base. Dada la ubicación de los satélites la ubicación de los vehículos pueden ser calculadas por triangulación.

2.4.3 - Uso de la Información Geográfica en los Problemas de Ruteo

Todas las clase de Problemas de Ruteo resueltos con cualquier técnica, ya sean métodos heurísticos o exhaustivos, precisan de alguna forma contar con un mínimo de información como puede ser la ubicación geográfica de los elementos, distancias y costos entre ellos, dada una calle conocer que movimientos puedo realizar. Para esto se debe tener una base de información geográfica detallada con toda la información necesaria para la resolución del problema.

Un ejemplo de la información geográfica usada en los Problemas de Ruteo en el área de servicio son: cuando se trata de un ruteo entre ciudades, que implica largos tramos entre ciudades, necesita acceder a datos sobre las redes de carreteras. Sin embargo un Ruteo en ciudades, que implica la distribución de cargas entre áreas urbanas, necesita acceder a datos detallados sobre calles, sentidos, etc. de la ciudad.

La información básica que debe disponer una base de datos geográfica para soportar un Problema de Ruteo de Vehículos es:

- Ubicación precisa de los clientes.
- Distancia entre sitios.
- Tiempos de viaje entre sitios.
- La geografía de las regiones de entrega y las carreteras.
- Información detalladas de las rutas.

Es muy importante que esta información geográfica sea exacta, por dos razones básicas: la estimación de los tiempos y distancias de viaje deben ser exactas, ya que subestimar pequeños errores puede aumentar considerablemente la utilización de conductores y vehículos; y además a mayor exactitud mayor calidad de la solución.

De esto último surge la necesidad del uso de Sistemas de Información Geográfica.

Capítulo 3 - Sistemas de Información Geográfica (G.I.S.)

3.1 - ¿Qué es un G.I.S.?

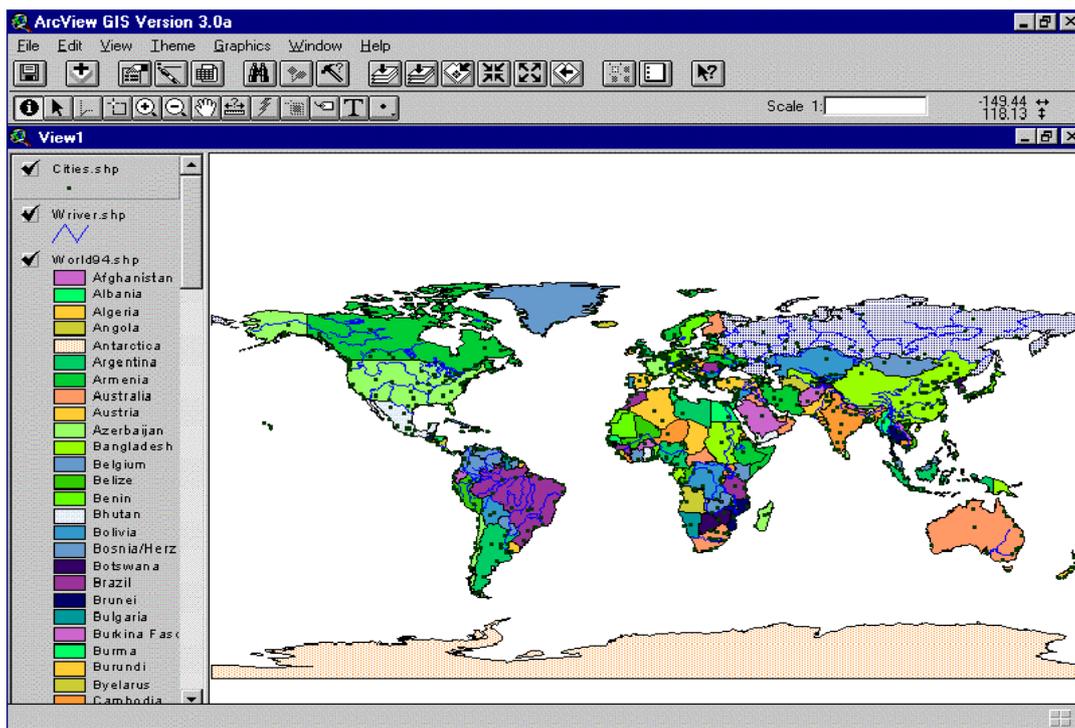
Los Sistemas de Información Geográfica (G.I.S.) son el avance más importante desde la invención del mapa para el manejo de datos espaciales. Un G.I.S. es un sistema que por medio de computadoras y datos geográficos ayuda a nuestro mejor entendimiento del mundo en que vivimos y nos permite resolver los problemas que diariamente afrontamos. A través de un G.I.S. los mapas pueden ser fácilmente integrados con otros datos, no simplemente almacena y despliega información acerca de lugares, sino que almacena y despliega información que puede ser relacionada con la ubicación geográfica de los mismos.

Los mapas G.I.S. cambian dinámicamente a medida que los datos asociados son actualizados, lo que hace que la información se mantenga fácilmente actualizada.

Un G.I.S. nos provee de las herramientas necesarias para analizar modelos, localizar eventos, medir cuán distantes están dichos eventos, encontrar la mejor manera de llegar a un destino y explorar como los problemas se relacionan con los demás.

Un G.I.S. se asemeja a un programa de bases de datos, ya que analiza y relaciona información almacenada bajo la forma de registros, con la diferencia de que cada uno contiene información usada para dibujar formas (puntos, líneas y polígonos).

Previo a la disponibilidad de la tecnología G.I.S., la forma en que se tomaban decisiones no siempre eran las más adecuadas. Se confiaba en mapas tradicionales y en tablas estadísticas impresas. Los mapas, las tablas y la cartografía eran difíciles de mantener actualizados. El resultado eran decisiones basadas en información pobre, solucionando solo parte del problema o simplemente realizando una mala planificación. Todas las alternativas no podían ser tenidas en cuenta ya que no podían ser visualizadas en conjunto. Adicionalmente, las alternativas que se consideraban eran generalmente basadas en datos incompletos.



3.2 - Origen y evolución de los Sistemas de Información Geográfica.

Durante mucho tiempo la recolección de datos acerca de la distribución espacial de las propiedades importantes de la superficie de la Tierra, ha sido una parte significativa de las actividades de las sociedades organizadas. Desde los comienzos de la civilización hasta los tiempos modernos, los datos espaciales han sido recolectados por navegantes y geógrafos, los cuales eran analizados y traducidos en forma de imágenes por los cartógrafos.

Originalmente los mapas fueron usados para describir lugares lejanos, como una ayuda para la navegación y la estrategia militar. El desarrollo en el entendimiento de los Recursos Naturales como Geología, Geomorfología y Ecología que comenzaron en el siglo XIX y continuaron hasta nuestros días, otorgaron nuevo material para ser mapeado.

La necesidad de datos y análisis espacial no ha tenido límites para los científicos del suelo. Planos urbanos y agencias catastrales necesitan información detallada acerca de la distribución de tierras y recursos en pueblos y ciudades.

La base de datos estaba representada por los mapas en papel y su información asociada, por lo que había varias consecuencias importantes para el almacenamiento, codificación, y uso de esa información.

Primero, el dato original tenía que ser reducido en volumen, o clasificado de manera de hacerlo entendible y representable; como consecuencia, varios detalles locales eran a menudo perdidos.

Segundo, el mapa tenía que ser dibujado con exactitud y la presentación, particularmente de temas complejos, tenía que ser muy clara.

Tercero, si se tenía un gran volumen de información, provocaba que las áreas que eran grandes con respecto a la escala del mapa fueran representadas por medio de varias hojas. Solía ocurrir que un área de interés estuviera en la intersección de dos hojas.

Cuarto, era difícil recuperar un dato perteneciente al mapa para ser comparados con otros.

Quinto, el mapa impreso era un documento estático, por lo que para actualizarlo se debía redibujarlo, agregando la nueva información, lo cual era muy costoso y consumía mucho tiempo.

Cuando se comenzó a utilizar las computadoras para la creación de los mapas, se observaron las siguientes ventajas:

- Creación de mapas existentes en forma más rápida y económica, acorde a las necesidades específicas de los usuarios.
- Permitir experimentar con diferentes representaciones gráficas para los mismos datos.
- Fácil realización y actualización del mapa.
- Minimizar el uso de mapas impresos como forma de almacenar los datos y los efectos de clasificación y generalización en la calidad de los datos.
- Creación de mapas que eran difíciles de realizar manualmente.
- Revisión de todo el proceso de creación del mapa lo cual trae aparejado ahorros y mejoras.

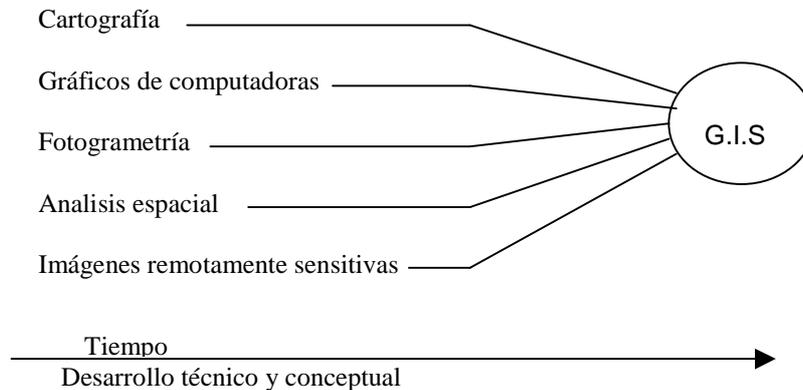
La introducción de la Cartografía asistida por computadora no llevó inmediatamente a un ahorro en el costo como era de esperarse. La adquisición y desarrollo de las nuevas herramientas fueron extremadamente costosas y llevaron un tiempo considerable de entrenamiento.

El mercado de mapeo asistido por computadora apuntaba a resolver un problema en particular para un mercado que tenía distintas necesidades; consecuentemente, muchos compradores de sistemas costosos fueron forzados a contratar un staff de programadores para adaptar el sistema a su situación real.

La historia del uso de computadoras para mapeo y análisis espaciales muestra que ha habido un desarrollo paralelo en la captura automática de datos, el análisis de datos y la presentación en varios campos relacionados ampliamente. Estos campos son mapeo topográfico, cartografía temática, ingeniería civil, geografía, ciencia de los suelos, fotogrametría, planificación rural y urbana, utilización de redes y análisis de imágenes. Consecuentemente, ha habido mucho esfuerzo duplicado; esta duplicación, en inicialmente separados pero muy relacionados campos, está actualmente resultando en la unión de varias clases de procesos de datos espaciales junto a verdaderos Sistemas de Información Geográfica de propósito general.

Informe Taller V

Esencialmente, todas estas disciplinas apuntan a la misma clase de operaciones, desarrollar un poderoso conjunto de herramientas para juntar, almacenar, recuperar, transformar, y desplegar datos espaciales del mundo real para un conjunto particular de propósitos. Este conjunto de herramientas constituye un **Sistema de Información Geográfica (G.I.S.)**.



Los datos geográficos describen objetos del mundo real en términos de:

- Su posición con respecto a un sistema de coordenadas conocido.
- Sus atributos no relacionados con su posición (color, costo, PH, etc.).
- Su interrelación espacial con otros que describe cómo ellos están unidos o cómo se pueden navegar a través de ellos.

Los Sistemas de Información Geográfica difieren de los gráficos de computadoras porque estos últimos solo despliegan y manipulan material visible, no prestan mucha atención a los atributos no gráficos que la entidad visible tenga o no y qué útil puede resultar el análisis de esos datos. Buenos gráficos de computadoras son esenciales para los Sistemas de Información Geográfica modernos, pero un paquete gráfico no es por sí solo suficiente para realizar las tareas esperadas.

Los Sistemas de Información Geográfica tienen mucho en común con los sistemas CAD. Ambos sistemas necesitan ser capaces de relacionar objetos con un marco de referencia, manejar atributos no gráficos y ser capaces de describir relaciones topológicas. Su principal diferencia es la cantidad de volumen, la diversidad de datos ingresados a un Sistema de Información Geográfica y la naturaleza especializada de los métodos de análisis usados.

Los Sistemas de Información Geográfica deben ser pensados para ser mucho más que un sistema que codifique, almacena y recupere datos acerca de aspectos de la superficie terrestre. Los datos en un Sistema de Información Geográfica deben ser como la representación de un modelo del mundo real, ya que estos datos pueden ser accedidos, transformados y manipulados interactivamente en un Sistema de Información Geográfica, pueden servir para el estudio de los procesos ambientales o la anticipación de los posibles resultados de las decisiones de planificación.

El diseñador de un Sistema de Información Geográfica debe poder contestar al usuario un número ilimitado de preguntas usando una cierta combinación de recuperación de datos y opciones de transformación. Algunas de estas preguntas generales son:

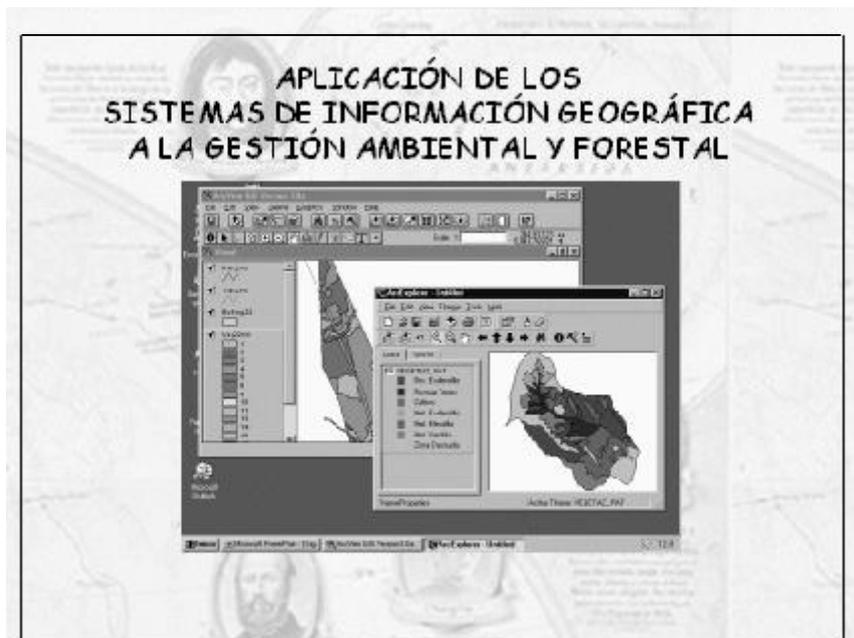
- ¿Dónde está el objeto A?
- ¿Dónde está A en relación a B?
- ¿Cuántas ocurrencias del tipo A hay a una distancia B de D?
- ¿Cuál es el valor de la función Z en la posición X?
- ¿Cuán grande es B (área, perímetro)?
- ¿Cuál es el resultado de interceptar varias clases de datos espaciales?
- ¿Cuál es el camino de menor costo, resistencia, distancia a lo largo del camino de X a Y pasando por P?

- ¿Cuáles son los objetos más cercanos a objetos que tienen cierta combinación de atributos?

3.3 - Aplicaciones de los Sistemas de Información Geográfica.

El uso de métodos y técnicas en estos sistemas, como herramientas idóneas para tratar esa información, se extienden en multitud de ámbitos. Así se han establecidos agrupaciones de aplicaciones en base a la similitud de temas que trata. Tal clasificación se trata de una aproximación pues sabemos que diariamente aparecen nuevas aplicaciones de los G.I.S. y que en el futuro continuarán apareciendo aún en mayor número.

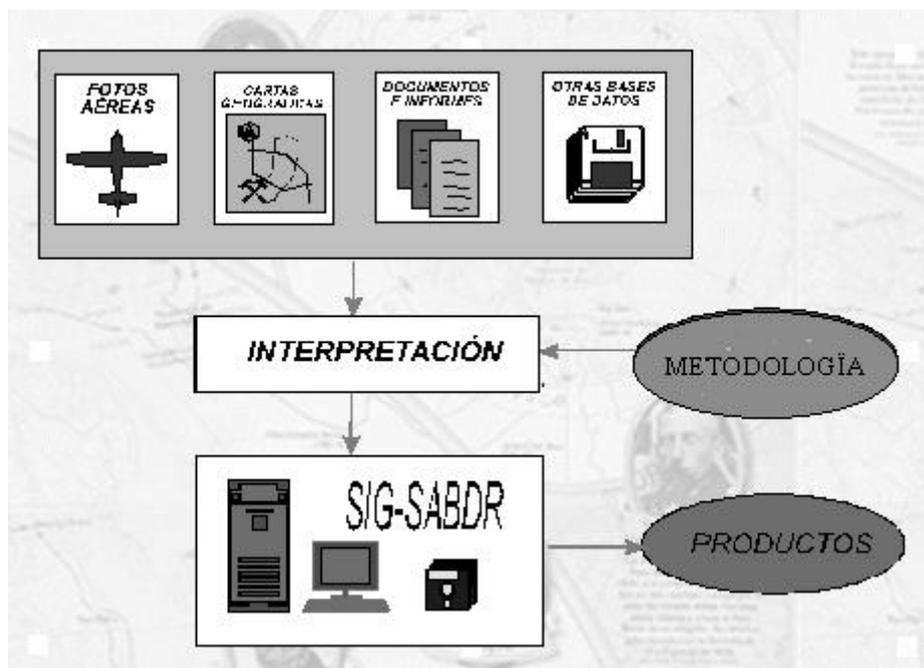
| | |
|--|--|
| Aplicaciones bióticas | Agricultura y usos del suelo. Gestión de los recursos naturales. Catastro. Planificación y gestión de servicios públicos. |
| Aplicaciones de Administración y Gestión. | Aplicaciones de carácter urbano. Aplicaciones cartográficas. Defensa y seguridad. |
| Aplicaciones socioeconómicas. | Censos y estadísticas de población. Análisis de mercado. CORINE (Coordinated Information on the European Environment Program). |
| Aplicaciones de carácter global. | Base de datos mundiales (Internet-Internetwork System - ARPANET) |



3.4 - Ventajas de usar G.I.S.

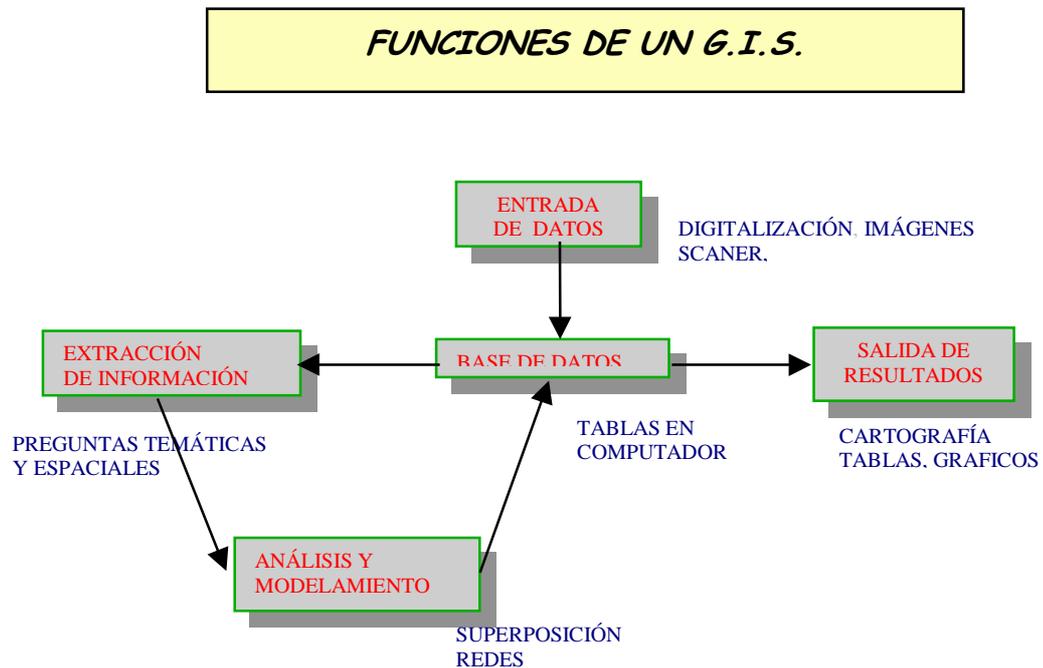
Por regla general, sabemos que la posibilidad de poder acceder a la información adecuada en el momento preciso significa tener una clara ventaja comparativa respecto a aquellos individuos que no tienen la capacidad de acceder a ella. Es por ello que la información a pasado a ser un recurso más de nuestras economías, siendo cada vez más necesaria para poder obtener bienes y proporcionar servicios. [Huxhold, 1992]. Por otro lado, es evidente para todos nosotros que las actividades humanas se han desarrollado a lo largo de siglos sobre la superficie terrestre, que aparece como continente y marco de sustentación sobre el cual el hombre actúa. Es por este motivo que el control sobre el territorio y, por lo tanto, sobre la información referente a él ha sido considerado siempre un factor clave [Goertz y Diehl, 1992].

Una de las ventajas de G.I.S. es que permite realizar el análisis de una gran cantidad de datos en corto tiempo ya que están almacenados en Bases de Datos, haciendo fácil su manejo mediante tablas, mapas, etc. Además permite reflejar la realidad, y cualquier cambio sobre ella puede ser visto casi al instante. Permite mostrar datos ingresados y resultados producto del análisis espacial, este producto son planos, mapas, tablas y gráficos. Además permite extraer y reorganizar la información de la Base de Datos, esto se realiza por medio de búsquedas temáticas y espaciales.



Otra ventaja es que los datos son mantenidos en formatos compactos pudiendo ser mantenidos y extraídos a muy bajo costo, pueden ser recuperados a gran velocidad, permitiendo realizar una variada serie de aplicaciones computacionales sobre ellos. También es posible manipular la información gráfica y no gráfica, realizar análisis espacial, verificar cambios temporales de los datos con gran velocidad y realiza producción cartográfica a gran escala y a bajos costos de manejo.

Es por todas las razones expuestas anteriormente que se implementó la estructura en un Sistema de Información Geográfico, ya que nos permite obtener fácil y rápidamente los objetos geográficos y la relación existente entre ellos.



3.5 - Los Sistemas de Información Geográfica hoy en día

Los campos de estudio de Sistemas de Información Geográfica son muy extensos y su desarrollo continúa a una tasa muy acelerada, a medida que surgen nuevas ideas y métodos.

Varias universidades alrededor del mundo ofrecen una variada gama de posgrados basados en esta área.

Actualmente existe una gran variedad de productos e información sobre este tema.

Los Sistemas de Información Geográfica son pues, sistemas para capturar, almacenar, chequear, analizar, y desplegar datos, los cuales están espacialmente referenciados a la tierra.

Los principales cambios con los cuales se encuentra el mundo hoy en día, como sobrepoblación, contaminación, deforestación, desastres naturales, tienen una dimensión geográfica crítica. Esto ha impulsado el desarrollo de Sistemas de Información Geográfica.

Los Sistemas de Información Geográfica otorgan la habilidad de crear mapas, integrar información, visualizar escenarios, resolver problemas complejos, presentar ideas y desarrollar soluciones efectivas.

Capítulo 4 - Estructura de Datos.

4.1 - ¿ Por que una estructura de datos?

Los Problemas de Ruteo tienen dos componentes importantes, la estructura de datos sobre la cual se corren los algoritmos, y los algoritmos en sí. Sin embargo actualmente en el mercado no hay una estructura de datos que soporte cualquier tipo de Ruteo de Vehículos, cada algoritmo en particular tiene por lo general asociada una estructura que se adecua al mismo de la mejor manera, pero cualquier cambio en dicho algoritmo hace que la estructura quede obsoleta o que haya que volver a diseñarla, por lo que se vuelve necesario tener una estructura de datos lo bastante genérica para que pueda soportar una amplia gama de este tipo de problemas. Por lo expresado antes el objetivo de nuestro Taller V, es diseñar e implementar una estructura de datos con sus correspondientes funciones de manejo, que permita cargar sobre la misma todos los componentes de un "Problema de Ruteo de Vehículos", resolverlo y recuperar los resultados.

4.2 - Esquematización de elementos para la construcción de la estructura.

Para alcanzar el objetivo de este Taller V se realizó además de la estructura de datos una interfase para que el usuario pueda manejarla de manera sencilla y amigable.

Para el desarrollo del Taller V se dispuso de los siguientes elementos y sus relaciones: la entrada, de donde se obtuvieron los datos con que se cargó la estructura y cuyo formato fue dado por el usuario; la estructura, la interfase para ella, los algoritmos para VRP utilizados para el testeo y las salidas, esto se ve reflejado en la Figura 3.

La entrada de datos para la carga de la estructura se obtiene de shapefiles². Estos shapefiles pueden ser de puntos, polilíneas o polígonos, en nuestro caso la entrada solo es de shapefiles de puntos o polilíneas. Los puntos son los objetos que representan a los clientes y los depósitos; y las polilíneas forman la red donde se realiza el ruteo.

Cada uno de estos shapefiles está formado entre otras cosas por archivos de base de datos .dbf. Los archivos .dbf están compuestos por campos con nombres particulares para cada shapefile. Para tener un significado y manejo estándar de cada uno de los campos, se dispone también, como entrada para cada shapefile, de un archivo con el mismo nombre y extensión .inf, que especifica el mapeo de estos campos. Un ejemplo sería: el campo "Peso1" que se puede corresponder con el largo de la calle para un shapefile o con la densidad del tránsito para otra. Además de estas entradas se tienen archivos de texto que contienen las reglas de tránsito, donde se indica entre que calles del mismo shapefile hay un costo asociado para doblar, estos archivos tienen el mismo nombre del shapefile y extensión .rgl. Para el manejo de vehículos se tiene una tabla y un archivo de texto con extensión .inf asociado con la información correspondiente para el mapeo con la tabla.

Basándonos en estas entradas, realizamos una estructura de datos genérica y una interfase para un manejo lo más transparente posible de la misma. La interfase se realizó mediante scripts en Avenue. Sobre esta estructura se programan los diferentes algoritmos de ruteo de una manera fácil, la salida de estos algoritmos son shapefiles y estructuras auxiliares para las que también se realizó una interfase. Para determinar si el objetivo de facilitar la tarea del programador se había cumplido, realizamos los algoritmos del Problema del Vendedor Viajero y Asignación de clientes a depósitos. Logramos el objetivo pero con el inconveniente de que la performance obtenida fue muy por debajo de la esperada, debido a que la que se obtiene con otros lenguajes de programación, por ejemplo: C++, es superior.

² Formato estándar de datos usado para soportar estructuras geográficas. [3]

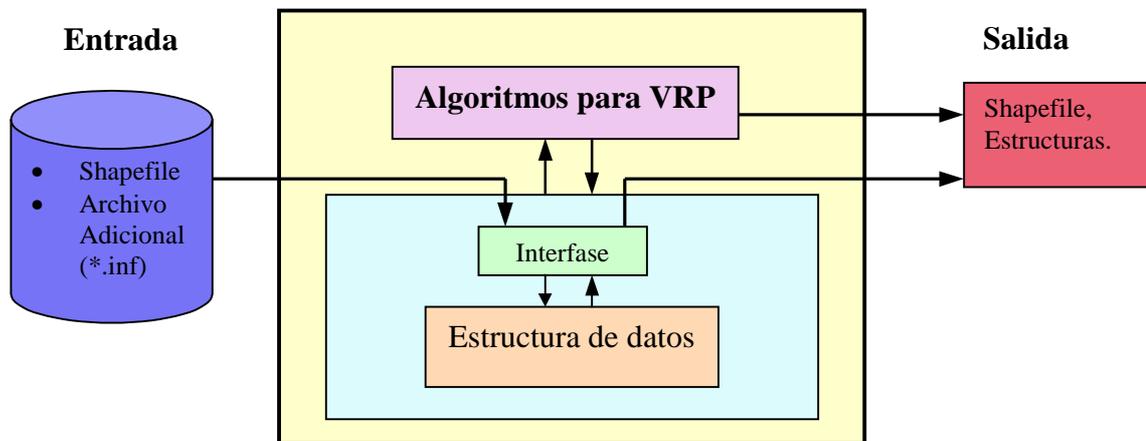


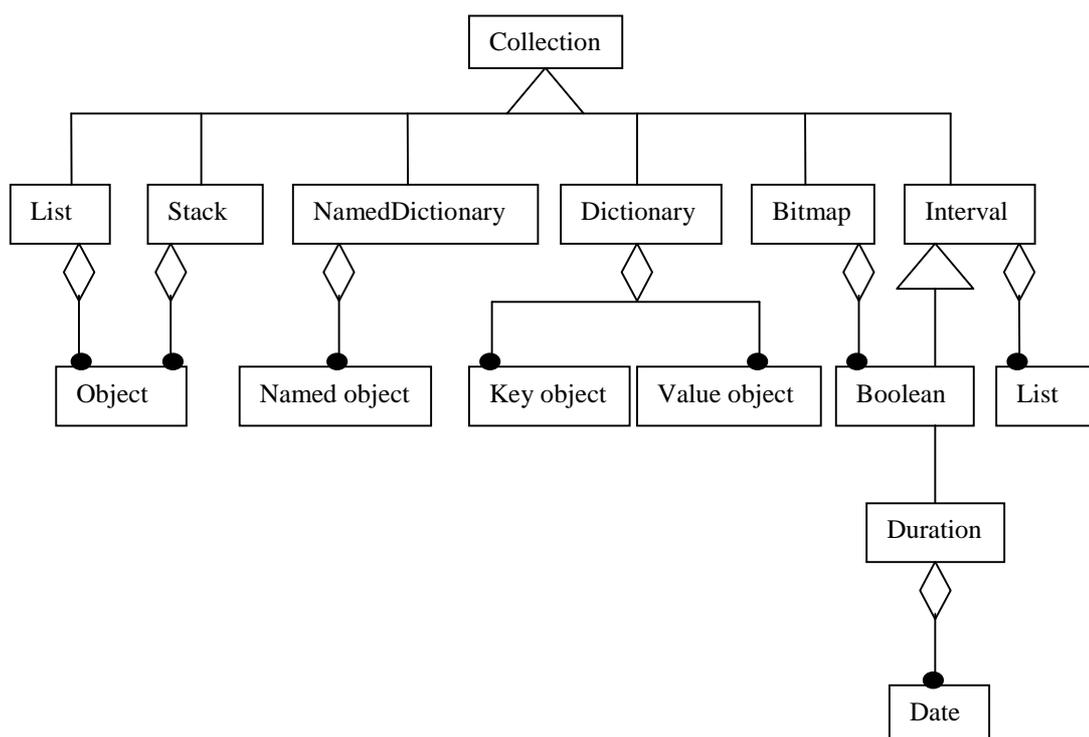
Figura 3: Diagrama de entradas y salidas:

4.3 - Implementación de la estructura.

La estructura se realiza sobre el lenguaje Avenue que es el lenguaje disponible para la herramienta ArcView, el cual es un Sistema de Información Geográfica (G.I.S.). Esta herramienta fue utilizada por requerimiento del usuario - Ing. José Comas.

Avenue es un lenguaje de programación orientado a objetos, el cual no dispone de un amplio conjunto de objetos de datos. Este lenguaje nos permite aprovechar las ventajas del uso de nociones geográficas pero nos dificulta la construcción de la estructura descrita anteriormente por la falta de objetos.

Para llegar a la decisión de cual tipo de datos utilizar, realizamos un estudio de los distintos objetos disponibles para colecciones. Estas clases de objetos son los presentados en el siguiente diagrama:



De este conjunto de colecciones nuestra decisión fue utilizar los diccionarios. Un diccionario es una colección desordenada de asociaciones entre objetos y valores, donde cada asociación es un elemento en el diccionario. Tanto la clave como el valor son objetos arbitrarios pero la clave solamente identifica el valor, o sea que una clave no debe ser usada en asociación a más de un valor. Este tipo de estructura de datos es eficiente cuando se necesita acceso randómico a un número de objetos. Los diccionarios en Avenue funcionan internamente calculando un número para cada clave, este número se usa como un índice dentro de un array de listas, cada lista contiene todos los valores cuya clave retorna el mismo número, esta tabla es una tabla Hash y las listas son listas de colisión. Para obtener la mayor performance de estos diccionarios al insertar datos en la estructura calculamos el porcentaje usado de la tabla del Hash y agregamos tamaño a la misma cuando excede cierto porcentaje.

Para la estructura se implementaron funciones de inserción, modificación, búsqueda, borrado y visualización de cuatro diccionarios enlazados. Estas estructuras y funciones se usan en la carga de puntos y de líneas.

Las estructuras creadas son:

- **_PtoInf:** esta estructura almacena toda la información necesaria de los archivos de información. Estos tienen el mapeo entre los nombres internos de los datos y los campos reales del shapefile. Es un enlace de dos diccionarios.
- **_shppto:** esta estructura almacena toda la información necesaria de los shapefile de puntos. Es un enlace de tres diccionarios.
- **_LinInf:** esta estructura almacena toda la información necesaria de los archivos de información. Estos tienen el mapeo entre los nombres internos de los datos y los campos reales del shapefile. Es un enlace de dos diccionarios.
- **_shplin:** esta estructura almacena toda la información necesaria de los shapefiles de líneas. Es un enlace de tres diccionarios.
- **_vehInf:** esta estructura almacena toda la información necesaria de los archivos de información. Estos tienen el mapeo entre los nombres internos de los datos y los campos reales de la tabla. Es un enlace de dos diccionarios.
- **_vehstruct:** esta estructura almacena toda la información correspondiente a los vehículos que se disponen. Es un enlace de tres diccionarios.
- **_reglas:** esta estructura almacena las reglas de tránsito de todos los shapefiles. Es un enlace de tres diccionarios.
- **_flags:** esta estructura almacena las flags virtuales que se van creando. Es un enlace de dos diccionarios.

4.4 - Las estructuras.

Las estructuras diseñadas son una para puntos, una para líneas, una para vehículos y por último una para las reglas de tránsito.

La estructura de puntos permite reflejar los clientes y los depósitos existentes en la red; y la de líneas refleja la red; estas estructuras se cargan desde shapefiles de puntos y líneas respectivamente que se reciben como entrada. Para ambas se cuenta además con estructuras auxiliares que sirven para la estandarización de los nombres de los atributos para los diferentes objetos. Dichas estructuras se cargan desde archivos de texto que también son recibidos como entrada.

La estructura de vehículos se carga desde una tabla de datos. Se cuenta también con un archivo de texto para la estandarización de sus atributos.

Finalmente la estructura para las reglas de tránsito modela el costo asociado a doblar desde una calle a otra dentro del mismo shapefile, y se carga a partir de un archivo de texto.

Para la implementación de algoritmos sobre la estructura se pueden agregar nuevos campos denominados flags virtuales, los cuales están asociadas a un shapefile en particular. Un ejemplo de estas flags virtuales puede ser el campo Visitado o Asignado para los algoritmos T.S.P y Asignación de clientes a depósitos respectivamente. La diferencia de estos campos con los ya cargados en la estructura es que no se cargan desde un shapefile sino que son cargados con un valor por defecto.

4.4.1. Estructura de Puntos.

_PtoInfs

Esta estructura guarda la información obtenida en los archivos de texto cuyo nombre es igual al del shapefile de puntos correspondiente y extensión .inf. Estos archivos contienen la información para el mapeo de los nombres de los campos internos y los nombres de los campos de los shapefiles de puntos.

Esta estructura es un diccionario de segundo nivel con el siguiente formato:

Clave: Nombre del shapefile

Valor: Diccionario con Clave: Nombre del campo interno. Ej: IdUsuario

Valor: Nombre del campo que debe buscar en el .dbf

Por ejemplo, para los shapefiles clientes.shp y depositos.shp, existen los archivos clientes.inf y depositos.inf respectivamente con el siguiente formato:

Clientes.inf

```
[IdUsuario]
[IdInterno]
IdCliente
[Demanda 1]
Demanda
[Demanda 2]
[Flag1]
[Flag2]
[Nombre]
Cliente
```

Depositos.inf

```
[IdUsuario]
[IdInterno]
IdDeposito
[Capacidad 1]
Capacidad
[Capacidad 2]
[Flag1]
[Flag2]
[Nombre]
Deposito
```

Por lo tanto, _PtoInfs será un diccionario de dos niveles con la siguiente información, como se muestra en la Figura 4.

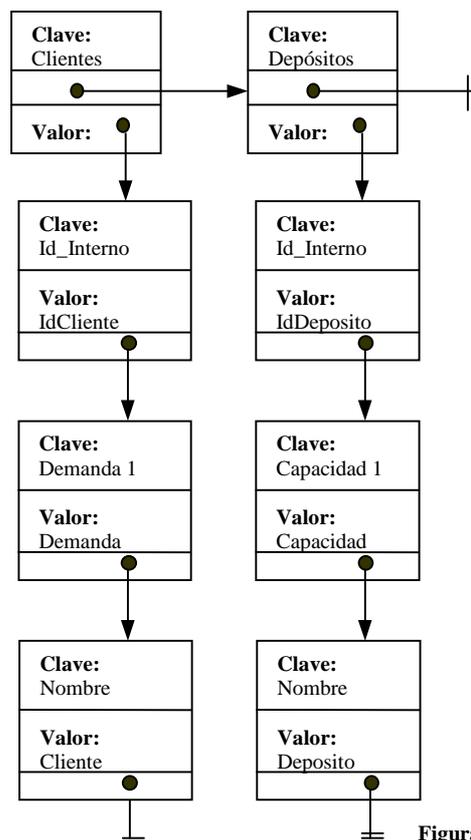


Figura 4: Ejemplo de la estructura de mapeo para puntos.

Informe Taller V

Esto significa que el shapefile de clientes tendrá un archivo .dbf con columnas de nombre IdCliente, Demanda y Cliente, y el shapefile de depósitos tendrá un dbf con columnas IdDeposito, Capacidad y Deposito.

Informe Taller V

_shpPto

Esta estructura guarda la información obtenida de los shapefiles de puntos. La información que se guarda es la que se obtiene del mapeo entre los campos reales del shapefile y los campos internos que están guardados en la estructura _PtoInfs.

Los campos internos son: IdInterno, IdUsuario, Demanda/Capacidad inicial 1, Demanda/Capacidad Inicial 2, ..., Demanda/Capacidad inicial N, Flag1, Flag2, ..., FlagN, Nombre. Estos campos pueden estar o no mapeados en el .inf y se cargan solo si lo están, con la excepción del IdInterno el cual es obligatorio. Se agrega además para el manejo interno las coordenadas corregidas del punto, luego de hacer (mediante cuentas) las correcciones de los errores de digitalización, acercándolo a la recta más cercana dependiendo de un radio dado. Otros datos que se ingresan son el identificador de la recta y el nombre del shapefile al cual esta pertenece, sobre la cual fue aproximado el punto, estos campos son PtoVdo, PtoVdoRct, PtoVdoShp respectivamente.

Además de estos campos existen operaciones para crear nuevos campos a los que llamamos "Flags Virtuales" y cargar los valores que se deseen. Estas flags se pueden eliminar, consultar y modificar.

Esta estructura es un diccionario de tercer nivel con el siguiente formato:

Clave: Nombre del shapefile

Valor: Diccionario con Clave: Valor del Identificador Interno.

Valor: Clave: Diccionario con Clave: Nombre de los campos que tiene asociado el Id.Interno.

Valor: Valor que tiene ese campo.

Clientes.dbf:

| IdCliente | Demanda | Cliente |
|-----------|---------|---------|
| 1 | 5 | Juan |
| 2 | 10 | Pedro |

Por lo tanto, _shpPto será un diccionario de tercer nivel como se muestra en la figura 5, teniendo en cuenta la estructura de mapeo asociada a él (ver Figura 4).

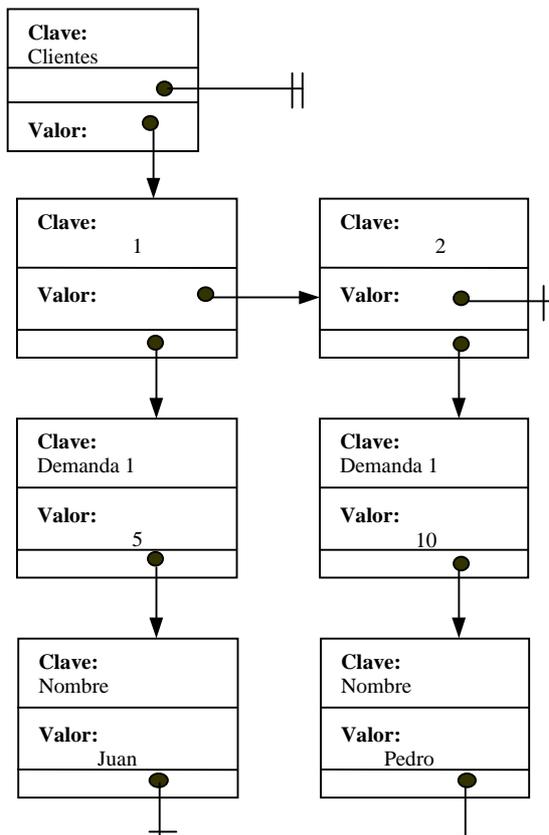


Figura 5: Ejemplo de la estructura de puntos.

_LinInfs

Esta estructura guarda la información obtenida en los archivos de texto cuyo nombre es igual al shapefile de líneas correspondiente y extensión .inf. Estos archivos contienen la información para el mapeo de los nombres de los campos internos y los nombres de los campos de los shapefiles de líneas.

Esta estructura es un diccionario de segundo nivel con el siguiente formato:

Clave: Nombre del shapefile

Valor: Diccionario con Clave: Nombre del campo interno. Ej: IdUsuario

Valor: Nombre del campo que debe buscar en el .dbf

Por ejemplo, para los shapefile de líneas llamados calles.shp y rutas.shp, existen los archivos calles.inf y rutas.inf respectivamente con el siguiente formato:

Calles.inf

```
[IdUsuario]
[IdInterno]
Street_
[Flecha]
[Habilitado]
[Peso1]
Peso1
[Peso2]
[Flag1]
[Nombre]
Street
```

Rutas.inf

```
[IdUsuario]
[IdInterno]
Route_
[Flecha]
[Habilitado]
[Peso1]
Peso1
[Peso2]
[Flag1]
[Nombre]
Route
```

Por lo tanto, _linInfs será un diccionario de segundo nivel con la siguiente información, como se muestra en la Figura 6.

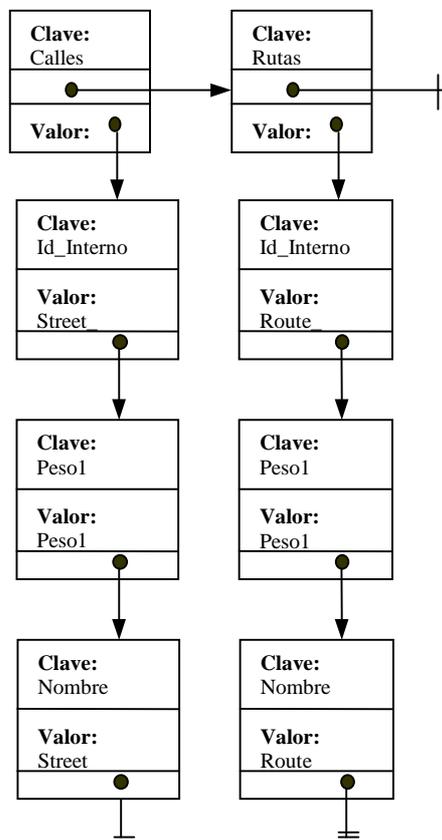


Figura 6: Ejemplo de la estructura de mapeo para líneas.

Informe Taller V

Esto significa que el shapefiles calles tendrá un archivo .dbf con columnas de nombre Street_, Peso1 y Street, y el shapefile rutas tendrá un dbf con columnas Route_, Peso1 y Route.

_shpLin

Esta estructura guarda la información obtenida de los shapefiles de líneas. La información que se guarda es la que se obtiene del mapeo entre los campos reales del shapefile y los campos internos que están guardados en la estructura `_LinInfs`.

Los campos internos son: `IdInterno`, `IdUsuario`, `Flecha`, `Habilitado`, `Peso1`, `Peso2`,...`PesoN`, `Flag1`, `Flag2`, `FlagN`, `Nombre`. Estos campos pueden estar o no mapeados en el `.inf` y se cargan solo si lo están, con la excepción del `IdInterno` el cual es obligatorio (pues es la clave2 dentro de la estructura). Se agrega además para un manejo interno de la estructura, las coordenadas de origen y destino de las líneas, estos campos son `PtoO` y `PtoD`.

Además de estos campos existen operaciones para crear nuevos campos a los que llamamos "Flags Virtuales" y cargar los valores que se deseen. Estas flags se pueden eliminar, consultar y modificar.

Esta estructura es un diccionario de tercer nivel con el siguiente formato:

Clave: Nombre del shapefile

Valor: Diccionario con Clave: Valor del Identificador Interno.

Valor: Diccionario con Clave: Nombre de los campos que tiene asociado el `Id.Interno`.

Valor: Valor que tiene ese campo.

Un ejemplo de esta estructura es:

Calles.dbf:

| Street_ | Peso1 | Street |
|---------|-------|---------|
| 1 | 18 | POLO |
| 2 | 15 | BEVERLY |

Por lo tanto, `_shpLin` será un diccionario de tercer nivel, como se muestra en la Figura 7, teniendo en cuenta la estructura de mapeo asociada a él (ver Figura 6).

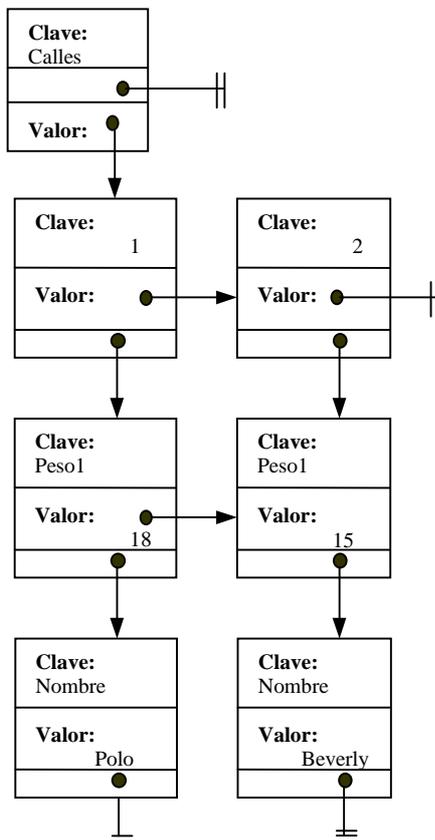


Figura 7: Ejemplo de la estructura de líneas.

_VehInfs

Esta estructura guarda la información obtenida en los archivos de texto cuyo nombre es igual a la tabla de vehículos correspondiente y extensión .inf. Estos archivos contienen la información para el mapeo de los nombres de los campos internos y los nombres de los campos de la tabla.

Esta estructura es un diccionario de segundo nivel con el siguiente formato:

Clave: Nombre de la tabla

Valor: Diccionario con Clave: Nombre del campo interno. Ej: IdUsuario

Valor: Nombre del campo con lo debe de buscar en el .dbf

Por ejemplo, para la tabla vehiculos.dbf , existe el archivo vehículos.inf con el siguiente formato:

Vehiculos.inf

```
[ IdInterno ]
ID
[ Capacidad 1 ]
Capacidad
[ Capacidad 2 ]
[TWmin]
[TWman]
[TiempoAtencion]
[ Flag1 ]
[ Nombre ]
Nombre
```

Por lo tanto, _VehInf será un diccionario de dos niveles, como se muestra en la Figura 8.

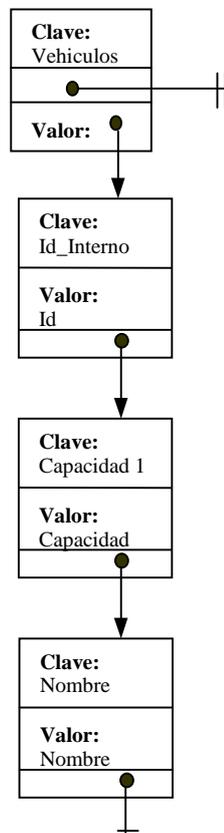


Figura 8: Ejemplo de la estructura de mapeo de vehículos.

Esto significa que la tabla de vehículos tendrá un archivo .dbf con columnas de nombre Id, Capacidad y Nombre.

_vehstruct

Esta estructura guarda la información obtenida de las tablas de vehículos. La información que se guarda es la que se obtiene del mapeo entre los campos reales de la tabla y los campos internos que están guardados en la estructura `_vehInfs`.

Los campos internos son: `IdInterno`, `IdUsuario`, `Capacidad 1`,...`Capacidad n`, `Twmin`, `Twman`, `TiempoAtencion`, `Flag1`, `Flag2`, `Flag3`, `Flag4`, `Flag5`, `Nombre`. Estos campos pueden estar o no mapeados en el `.inf` y se cargan solo si lo están, con la excepción del `IdInterno` el cual es obligatorio (pues es la clave2 dentro de la estructura).

Además de estos campos existen operaciones para crear nuevos campos a los que llamamos "Flags Virtuales" y cargar los valores que se deseen. Estas flags se pueden eliminar, consultar y modificar.

Esta estructura es un diccionario de tercer nivel con el siguiente formato:

Clave: Nombre de la tabla

Valor: Diccionario con Clave: Valor del Identificador Interno.

Valor: Diccionario con Clave: Nombre de los campos que tiene asociado el `Id.Interno`.

Valor: Valor que tiene ese campo.

Un ejemplo de esta estructura es:

Vehículos.dbf:

| ID | Capacidad | Nombre |
|----|-----------|----------|
| 1 | 10 | Camión 1 |
| 2 | 16 | Camión 2 |

Por lo tanto, `_vehstruct` será un diccionario de tercer nivel como se muestra en la Figura 9, teniendo en cuenta la estructura de mapeo asociada a él (ver Figura 8).

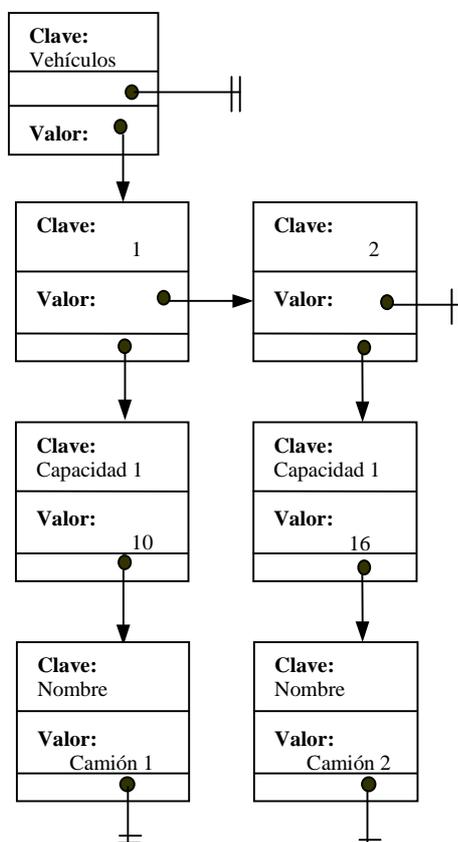


Figura 9: Ejemplo de la estructura para vehículos.

_reglas

Esta estructura guarda la información referente a las reglas de tránsito que tienen las calles. La información que se guarda es la que se obtiene del archivo que tiene el nombre del shapefile y extensión .rgl.

Esta estructura guarda la información necesaria para controlar lo siguiente: si se puede doblar de una calle a otra y con que costo. Si existe al menos una regla para una calle se agrega en la estructura _shplin el campo HayRegla y se carga con valor 1. La ausencia de reglas es que se puede doblar sin costo alguno. Una regla con valor -*_ significa que el costo es infinito y por lo tanto no se puede doblar.

Esta estructura es un diccionario de tercer nivel con el siguiente formato:

Clave: Nombre del shapefile

Valor: Diccionario con Clave: Identificador Interno de la calle origen.

Valor: Diccionario con Clave: Identificador Interno de la calle destino

Valor: Costo que tiene de ir de la calle origen a la calle destino.

Un ejemplo de esta estructura es:

Calles.rgl

| Id_arco_origen | Id_arco_destino | Costo |
|----------------|-----------------|-------|
| 239 | 240 | 1 |
| 239 | 301 | 40 |
| 296 | 293 | - *_- |

Por lo tanto, _regla será un diccionario de tercer nivel como se muestra en la Figura 10, teniendo en cuenta el archivo .rgl asociado a él.

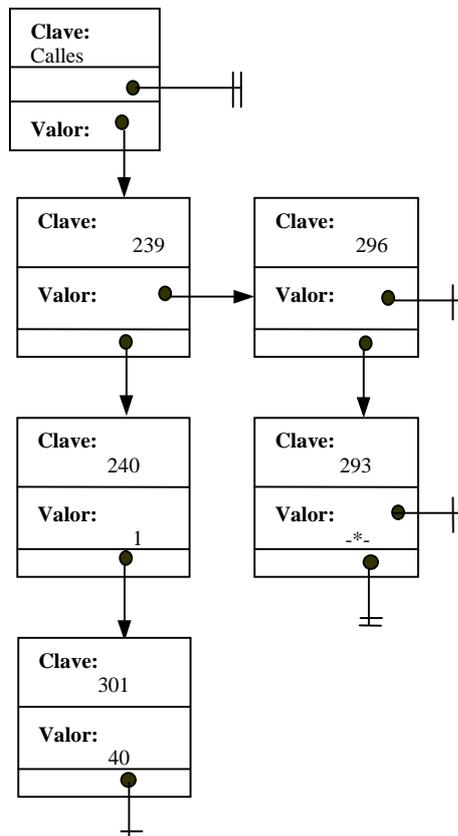


Figura 10: Ejemplo de la estructura de reglas de tránsito.

_flags

Esta estructura guarda las flags que se van creando para un shapefile considerado.

Esta estructura es un diccionario de segundo nivel con el siguiente formato:

Clave: Nombre del shapefile

Valor: Diccionario con Clave: Nombre de la flag.

Valor: Valor de la flag.

Un ejemplo de esta estructura es:

Si se quiere agregar una flag de visitado a un shapefile de clientes, con un valor por defecto, al agregarla se guarda la información en la estructura, como se muestra en la Figura 11.

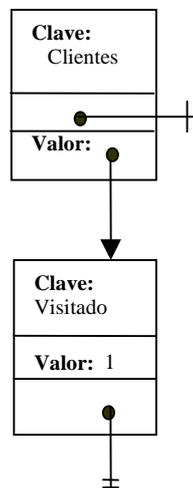


Figura 11: Ejemplo de una estructura de Flags Virtuales.

4.5 - Implementación en Avenue.

Realizar la implementación en Avenue nos brinda la posibilidad de manejar los objetos geográficos de forma transparente, y hacer análisis espacial de forma natural.

La principal dificultad o desventaja que presenta Avenue es que aunque es un lenguaje orientado a objetos no es posible crear nuevos objetos a partir de otros ya existentes. Además los objetos que brinda son muy pocos, lo cual dificulta la creación de estructuras de datos y hace menos performantes las operaciones sobre ella. Esto no hubiera ocurrido con otros lenguajes como por ejemplo C++, en el que se puede construir cualquier tipo de objetos sin inconvenientes pero hubiera dificultado el manejo de objetos geográficos.

Capítulo 5 - Resultados del testeo, tiempos de corrida.

El desarrollo de sistemas de software implica una serie de actividades de producción en las que las posibilidades de que aparezca la falibilidad humana son enormes. Los errores pueden empezar a darse desde el primer momento del proceso, en el que los objetivos pueden estar especificados de forma errónea o imperfecta, así como en los posteriores pasos de diseño y desarrollo.

La creciente inclusión del software como un elemento más de muchos sistemas y la importancia de los "costos" asociados a un fallo del mismo están motivando la creación de testeos minuciosos y bien planificados.

Los objetivos en nuestro testeo serán:

- Descubrir la mayor cantidad de errores.
- Crear casos de testeo que tengan alta probabilidad de mostrar un error que no se haya descubierto anteriormente con la mínima cantidad de esfuerzo y tiempo.

Cualquier producto de ingeniería puede ser probado de una de dos formas:

- Conociendo la función específica para la que fue diseñada el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa. Denominado Testeo de Caja Negra.
- Conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que "todas las piezas encajan", o sea que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. Denominado Testeo de Caja Blanca.

En el Testeo de Caja Negra se pretende demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Los errores que intentamos encontrar con este testeo son de la siguiente categoría:

- funciones incorrectas o ausentes.
- errores de interface.
- errores en la estructura de datos o accesos a las bases de datos externas.
- errores de rendimiento.
- errores de inicialización y de terminación.

Con el Testeo de Caja Blanca vamos a obtener casos de prueba que:

- garanticen que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo.
- ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas.
- ejecuten todos los bucles en sus límites y con sus límites operacionales.
- ejerciten las estructuras internas de datos para asegurar su validez.

Nuestro testeo está formado por las dos formas de testeo discutidas anteriormente.

La estrategia de testeo que utilizamos será en el siguiente orden:

- Testeo de unidad que hace uso intensivo de las técnicas de testeo de Caja Negra, ejecutando caminos específicos para asegurar un alcance completo y una detección máxima de errores.
- Testeo de integración que dirige todos los aspectos asociados con el doble problema de verificación y de construcción del programa. Durante la integración, las técnicas que más prevalecen son las de diseño de casos de prueba de la caja negra, aunque se pueden llevar a cabo unas pocas pruebas de la caja blanca con el fin de asegurar que se cubren los principales caminos de control.
- Testeo de Validación proporciona una seguridad final de que el software satisface todos los requisitos funcionales, de comportamiento y de

rendimiento. Durante la validación se usan exclusivamente técnicas de testeo de Caja Negra.

- Testeo del Sistema verifica que cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total.

5.1 - Tiempos de Corrida.

5.1.1. - Juegos de datos utilizados.

Los juegos de datos utilizados para obtener los tiempos de ejecución de los distintos algoritmos fueron:

Red 1: 47 calles. Trozo del mapa de Atlanta.

Red 2: 2107 calles. Mapa de Atlanta.

Red 3: 22216 calles. Mapa de Montevideo.

Clientes 1: 18 clientes con 4 campos cada uno.

Clientes 2: 52 clientes con 16 campos cada uno.

Clientes 3: 130 clientes.

Depósitos 1: 3 depósitos.

5.1.2. - Tiempos de corrida de la carga de la estructura.

Se controlaron los tiempos de corrida de la carga de todos los elementos necesarios para considerar los problemas de VRP, estos tiempos se obtuvieron según la cantidad de clientes, de calles y la cantidad de datos existentes para cada uno de ellos.

La carga de esta estructura utiliza las siguientes funciones:

T5.Crg_Inf_Lin: Carga el mapeo entre los campos internos de la estructura y los campos del. dbf del shapefile desde un archivo .inf de líneas .

T5.Crg_Inf_Pto: Carga el mapeo entre los campos internos de la estructura y los campos del. dbf del shapefile desde un archivo .inf de puntos .

T5.Crg_Struct_Lin: Carga en la estructura de líneas los datos que se marcaron como disponibles en el archivo .inf.

T5.Crg_Struct_Ptos: Carga en la estructura de puntos los datos que se marcaron como disponibles en el archivo .inf.

Los tiempos de ejecución fueron probados sobre los siguientes casos:

Caso 1: Red 1. Trozo de Atlanta.

Clientes 1.

El tiempo obtenido fue de 15".

Caso 2: Red 2. Atlanta

Clientes 2.

El tiempo obtenido fue de 1'50"

Caso 3: Red 3. Montevideo.

Clientes 3.

El tiempo obtenido fue de 30'18"

Informe Taller V

5.1.3. - Tiempos de corrida de los distintos algoritmos de Dijkstra.

T5.Dijkstra_ID_P : Dijkstra entre dos identificadores de puntos para ciertos shapefiles que para cuando encuentra el camino más corto entre ellos.

T5.Dijkstra_ID_NP : Dijkstra que encuentra el camino mas corto entre un identificadores de puntos para ciertos shapefiles y todos los demás.

Los tiempos de ejecución fueron probados sobre los siguientes casos, los cuales se diferencian por la cantidad de arcos existentes en la red, considerando el peor caso:

Caso 1: Red con 47 arcos. Trozo de Atlanta.

Caso 2: Red con 2107 arcos. Atlanta.

Caso 3: Red con 22216 arcos. Montevideo.

| Script | Caso 1. | Caso 2. | Caso 3. |
|-------------------|----------------|----------------|----------------|
| T5.Dijkstra_ID_P | 0'12" | 3'20" | 8:45' |
| T5.Dijkstra_ID_NP | 0'15" | 3'25" | **** |

5.1.4 - Tiempos de corrida de los algoritmos de TSP.

Los tiempos de ejecución fueron probados sobre los siguientes casos, los cuales se diferencian por la cantidad de arcos existentes en la red, considerando el peor caso:

Caso 1: Red con 47 arcos. Trozo de Atlanta.

Caso 2: Red con 2107 arcos. Atlanta.

Caso 3: Red con 22216 arcos. Montevideo.

| Caso 1. | Caso 2. | Caso 3. |
|----------------|----------------|----------------|
| 1'14" | 11:12' | ***** |

5.1.5. - Tiempo de corrida de los algoritmos de Asignación de clientes a depósito.

Los tiempos de ejecución fueron probados sobre los siguientes casos, los cuales se diferencian por la cantidad de arcos existentes en la red, considerando el peor caso:

Caso 1: Red con 47 arcos. Trozo de Atlanta.

Caso 2: Red con 2107 arcos. Atlanta.

Caso 3: Red con 22216 arcos. Montevideo.

| Caso 1. | Caso 2. | Caso 3. |
|----------------|----------------|----------------|
| 44" | 8:23' | **** |

Nota: Los testeos se realizaron en una máquina con las siguientes características:

Procesador: Pentium III 450 MHz

Memoria RAM: 64 Mb.

Capítulo 6 - Conclusiones y trabajo futuro.

Las conclusiones de este Taller V son las siguientes:

- Logramos el objetivo de este Taller V. Desarrollamos una Estructura de Datos genérica que permite solucionar cualquier tipo de VRP, evitándose la creación de una distinta para cada problema en particular. La estructura es buena porque es fácil de usar y sirve para cualquier tipo de VRP. El problema hallado es que los tiempos de ejecución obtenidos para la carga, manejo e implementación de algoritmos sobre ella no son buenos.
- Es novedoso, porque como se desprende de la búsqueda bibliográfica, hasta este momento no existían estructuras genéricas para solucionar Problemas de Ruteo de Vehículos, sino que había una estructura diseñada especialmente para cada problema particular, lo cual dificultaba la resolución de los algoritmos.
- El lenguaje de programación utilizado, Avenue, es muy bueno para trabajar con objetos geográficos, lo que nos permite hacer análisis espacial de forma natural, pero no es un lenguaje eficiente para crear una estructura genérica ya que aunque es orientado a objetos, no se pueden crear nuevos objetos a partir de los disponibles por el mismo. Los objetos que se disponen para la creación de la estructura son muy pocos, por lo que no se tuvo alternativa al elegirlos para el diseño de la misma.
- Claramente Avenue no está diseñado para generar nuevas estructuras, ampliar el lenguaje o correr sobre él algoritmos "pesados". Esto hace que los tiempos obtenidos no sean óptimos. Los tiempos de corrida que obtuvimos para algunos de los algoritmos que implementamos, ejecutados sobre Atlanta (2107 calles) fueron los siguientes:
 - Dijkstra de un punto a todos los restantes: 3'25"
 - Asignación de clientes a depósitos: 8:23'
 - TSP: 11:12'
- Los tiempos obtenidos en funciones que recorren la estructura y realizan una gran cantidad de cálculos en memoria, para lo que Avenue no es muy recomendable, no fueron óptimos. Un ejemplo de estas funciones es Dijkstra, por este motivo es recomendable que en trabajos futuros se implementen estas funciones en otros lenguajes, como por ejemplo C⁺⁺.
- Otros trabajos futuros, además del mencionado anteriormente son:
 - Poder asociar los clientes a un lado de las calles que conforman la red y no sobre ella como está implementado actualmente.
 - Poder considerar el costo asociado a cruzar las calles.
 - Agregar semáforos a la red.
 - Dentro de los requerimientos del usuario solo se consideró las reglas de tránsito asociadas al mismo shapefile, se podrían extender estas a reglas de tránsito entre calles de shapefiles distintos.
 - La estructura implementada no considera la movilidad de los vehículos sobre la red, por lo que se podría permitir en un futuro simular movimientos de estos, de manera de reproducir los movimientos de los móviles de una corrida, agregando todas las funciones que fueran necesarias para realizarlo.
 - Permitir la segmentación dinámica, es decir poder dividir la segmentación física en una segmentación lógica con distintos costos asociados y cambiar esta segmentación dinámicamente.

Bibliografía.

- [1] - Routing and Scheduling of Vehicles and Crews. Lawrence Bodin, Bruce Golden, Arjang Assad and Michael Ball.
- [2] - Modeling and Implementation Issues in Vehicle Routing. Arjang A. Assad.
- [3] - Esri.
http://www.esri.com/library/gis/abtgis/gis_do.html
- [4] - Network Engine.
- Principles of Geographical Information Systems for Land Resources Assessment
P.A. Burrough
- Ingeniería del Software. Un enfoque práctico. Roger S. Pressman.
- Sistemas de Información Geográfica.
<http://members.tripod.com/rioja/sig.htm>
- National Remote Sensing Centre (NRSC), UK
<http://www.ncl.ac.uk/ngraphic/wotzgis.html>
- I.C.A. –Ingenieros Consultores Asociados.
<http://www.ica.com.uy>
- What Can G.I.S. Do for You?
http://www.esri.com/library/gis/abtgis/gis_do.html
- What Is a G.I.S.?
http://www.esri.com/library/gis/abtgis/what_gis.html
- G.I.S in Everyday Life
http://www.esri.com/library/gis/abtgis/gis_life.html
- What is G.I.S.?
<http://www.ncl.ac.uk/~ngraphic/wotzgis.html>
- OR-NOTES
<http://mscmga.ms.ic.ac.uk/jeb/or/vrp.html>
- Vehicle Routing
<http://primal.iems.nwu.edu/~levi/route.html>
- A Location Based Heuristic for General Routing Problems
<http://primal.iems.nwu.edu/~levi/route.html>
- Vehicle Routing with Multiple Use of Vehicles
<http://primal.iems.nwu.edu/~levi/route.html>
- Probabilistic Analysis and Practical Algorithms for the Vehicle Routing Problems with Time Windows.
<http://primal.iems.nwu.edu/~levi/route.html>

Informe Taller V

- The Vehicle Routing Routing with Time Window (VRPTW)
<http://www.ens.fr/~caseau/vrptw.html>