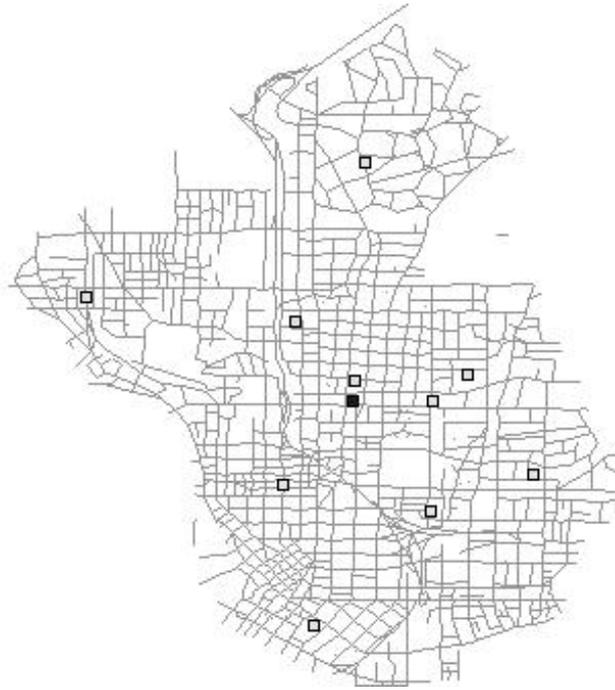


# RUTEO DE VEHÍCULOS



*Facultad de Ingeniería*

*Taller 5*

**Carlos Pedezert Nuñez**

**Javier Barreiro Piana**

**Nicolás Sosa Andriolo**

## Abstract

El presente informe refiere al desarrollo de un proyecto de software sobre “Ruteo de Vehículos” solicitado por la empresa I.C.A. S.A. y realizado como Taller 5 de Facultad de Ingeniería.

El objetivo es desarrollar un software que resuelva la mayor cantidad posible de problemas de ruteo, logrando de esta forma un paquete genérico de acuerdo a los requerimientos del usuario, los cuales incluyen los tipos de problemas a resolver. Además debe servir de base para futuros proyectos de casos más específicos. El interés por este tipo de producto surge pues en el mercado no existe ningún producto que resuelva diversos tipos de problemas de ruteo.

Vemos los distintos tipos de problemas de ruteo resueltos y las diferentes consideraciones a tener en cuenta a la hora de realizar un paquete genérico de Ruteo de Vehículos

El resultado es el desarrollo de un paquete de software genérico de acuerdo a los requerimientos del usuario. Dicho software resuelve los problemas básicos y las extensiones más significativas y puede ser expandido fácilmente para resolver problemas particulares.

## Indice

1	Introducción.....	4
2	Definición de problemas.....	7
	TSP.....	7
	MTSP.....	8
	VRP Clásico.....	8
	Extensiones del problema clásico.....	11
	CPP.....	13
	DARP.....	13
	Software comercial, VRP y extensiones.....	14
3	Requerimientos de Usuario.....	15
4	Implementación.....	16
	Arquitectura del Sistema y Lenguajes de Programación.....	16
	Elección de Algoritmos.....	16
	Interfase de usuario.....	17
	Estructuras de datos utilizadas para los algoritmos de ruteo.....	21
	VRP.....	22
	TSP.....	23
	MDVRP.....	23
	VRPTW.....	24
	MDVRPTW.....	24
	Flota Heterogénea.....	25
	CPP.....	26
	DARP.....	26
5	Testeo.....	28
6	Dificultades.....	30
	MapObjects.....	30
	Generación de dll.....	30
7	Resultados y conclusiones.....	32
8	Referencias.....	33
9	Apéndices.....	34
	Soluciones Existentes Actualmente.....	35
	Control De Reuniones.....	83
	Plan De Trabajo Y Avance Del Proyecto.....	87

# 1 Introducción

Este informe trata sobre el desarrollo de un proyecto de software sobre Ruteo de Vehículos solicitado por la empresa I.C.A. S.A. a través de su área de Optimización y realizado como formación académica por estudiantes de la Facultad de Ingeniería del Uruguay en el marco de la materia Taller 5. La propiedad intelectual del producto final es de la empresa I.C.A. S.A.

El problema clásico de Ruteo de Vehículos (VRP) se define como: “Determinar un conjunto de rutas de costo mínimo para una flota homogénea de vehículos que sirve a un conjunto de clientes dispersos geográficamente”. Este problema tiene las siguientes restricciones:

- Cada vehículo comienza y termina su ruta en un mismo depósito.
- Cada cliente es visitado exactamente una vez y por un único vehículo que satisface toda su demanda.
- Ninguna ruta excede la capacidad del vehículo que la recorre

El problema de Ruteo de Vehículos ha sido un área de creciente interés en las dos últimas décadas. El interés por resolver este problema surge tanto por los enormes costos que deben afrontar las empresas dedicadas a la distribución y/o recolección por concepto de fletes, como por la enorme complejidad asociada a la resolución de este tipo de problemas, el VRP pertenece a la clase de problemas denominados NP-duros [b].

Los costos asociados con la operación de vehículos y tripulaciones para propósitos de reparto forman un importante componente de los costos totales de distribución, aproximadamente un 16% del valor de venta de un producto[b]. Por lo tanto pequeños porcentajes de ahorro pueden deparar un importante ahorro en el gasto a lo largo del tiempo.

Las principales motivaciones para desarrollar un producto de software que resuelva los problemas de ruteo de vehículos que se presentan en nuestra vida diaria son:

- Disminución de costos por concepto de fletes, personal, mantenimiento, etc.
- Ahorro de tiempo.
- Obtener una mayor cantidad de soluciones alternativas.
- Mejorar el servicio al cliente.

Al coordinar una distribución eficiente de los productos de una empresa se presentan una variedad de problemas de decisión en el ámbito estratégico, táctico y operativo. Las decisiones relacionadas con la ubicación de plantas y/o depósitos se pueden ver como estratégicas, mientras que los problemas de tamaño y tipo de flota son decisiones tácticas. Finalmente, en el ámbito operativo, deben ser consideradas varias decisiones concernientes a las personas que van a manejar los vehículos. La distinción entre la planificación estratégica, táctica y operativa no debería ser interpretada en forma muy rígida, pues las decisiones involucradas están muy relacionadas entre sí.

Además de la ubicación de los depósitos, la planificación efectiva de repartos incluye otro tipo de decisiones que incluyen:

- Identificar el tamaño del área que cada depósito debe abastecer
- El tamaño y tipo de flota disponible en cada depósito
- Frecuencia de repartos a cada cliente

Dadas las decisiones descriptas anteriormente, la empresa debe rutear sus vehículos de forma que los costos sean mínimos.

Existe una gran variedad de problemas de Ruteo de Vehículos. Por ejemplo problemas con uno o varios depósitos, con o sin restricciones de tiempo, con flota de vehículos homogénea o heterogénea, etc.

En la siguiente tabla vemos las diferentes características de los problemas de ruteo y las opciones posibles de cada una.

CARACTERISTICAS	OPCIONES POSIBLES
Tamaño de la flota	Un vehículo Varios vehículos
Tipo de flota	Homogénea (sólo un tipo de vehículo) Heterogénea (distintos tipos de vehículos)
Almacenaje de vehículos	Depósito único Varios depósitos
Naturaleza de la demanda	Demanda determinística (conocida) Requerimientos estocásticos de demanda Satisfacción de demanda variable

Ubicación de la demanda	En los nodos En los arcos En ambos
Red de aristas	Unidireccional Bidireccional Aristas unidireccionales y aristas bidireccionales Euclidea
Restricciones de capacidad de los vehículos	Fija. Todos los vehículos la misma capacidad Fija. Capacidad variable de acuerdo al tipo de vehículo Sin restricciones de capacidad en los vehículos
Tiempo máximo de recorrido de una ruta	Fijo. El mismo tiempo para todas las rutas Fijo. Diferente tiempo para cada ruta Sin restricciones de tiempo para las rutas
Operaciones	Sólo pickups (carga) Sólo delivery (entrega) Pickup y delivery
Costos	Costos variables de ruteo. Costos fijos operativos y de compra de vehículos
Objetivos	Minimizar costos totales de ruteo Minimizar total de costos fijos y variables. Minimizar número de vehículos necesarios Maximizar función de utilidad basada en servicio y conveniencia Maximizar función de utilidad basada en prioridades del cliente

Los problemas de ruteo más simples son aquellos en que no existen restricciones temporales ni relaciones de precedencia. La salida de todos los sistemas de ruteo es básicamente la misma: A cada vehículo se le asigna una ruta y un horario. Generalmente, la ruta especifica la secuencia de lugares que deben ser visitados y el horario identifica las horas en que cada actividad debe llevarse a cabo en esos lugares.

El objetivo de este Taller es desarrollar un software que resuelva la mayor cantidad posible de problemas de ruteo, pues en el mercado nacional no existe ningún producto que resuelva todos los casos. Más adelante en este informe se presentan algunos productos existentes y su funcionalidad con lo que vemos que en el mercado internacional tampoco es común encontrar un software que resuelva gran variedad de problemas. Este software también servirá como punto de partida o referencia para la resolución de problemas concretos con restricciones específicas.

Podemos diferenciar dos enfoques a la hora de resolver este tipo de problemas. Se pueden utilizar algoritmos exactos los cuales nos llevan a la mejor solución o algoritmos heurísticos los cuales nos aproximan a la mejor solución. Los algoritmos exactos son aplicables sólo para problemas muy pequeños (50-100 clientes) por lo que utilizaremos algoritmos heurísticos ya que buscamos realizar un software que resuelva gran cantidad de casos lo cual implica además de variedad, la posibilidad de resolver casos de gran porte. Además los problemas tratados son NP-Complejos [b] lo que también nos lleva a utilizar algoritmos heurísticos.

Resultados: se desarrolla un paquete de software genérico de acuerdo a los requerimientos del usuario. Dicho software resuelve los problemas básicos y las extensiones más significativas y puede ser expandido fácilmente para resolver problemas particulares. En la siguiente tabla vemos algunos paquetes de software difundidos comercialmente y que casos resuelve cada uno de ellos.

	TSP	TSP TW	TSP FH	VRP	VRP TW	VRP FH	MDVRP	CPP	DARP
DynaRoute	-	-	-	-	-	-	-	-	Si
GeoRoute 5	Si	-	-						
GeoPostal	-	-	-	-	-	-	-	Si	-
RouteSmart	Si	-	-						
SHIPCONS II	Si	-	-						
TransCad	Si	-							
Trapeze-Taxi	-	-	-	-	-	-	-	-	Si
Direct Route	Si	Si	Si	Si	Si	Si	S/i	-	-
GeoRoute-Municipal	-	-	-	-	-	-	-	Si	-
LoadExpress Plus	Si	-	-						
RiMMS	Si	s/i	s/i	Si	s/i	s/i	S/i	-	-
RoutePro Dispatcher	Si	-	-						
Routronics 2000	-	-	-	-	-	-	-	-	Si
STARS	Si	-	-						
Optrak	Si	s/i	s/i	Si	s/i	s/i	Si	-	-
TruckStops for Windows	Si	-	-						
ArcLogistics Route	Si	-	-						
<b>Nuestro sistema</b>	<b>Si</b>								

#### ESTRUCTURA DEL INFORME:

- CAP 2 Definición de problemas a tratar. En este capítulo vemos con detenimiento la definición de los problemas que resolveremos. Vemos los casos básicos de ruteo de vehículos y las principales extensiones de dichos casos.
- CAP 3 Requerimientos de Usuario.
- CAP 4 Implementación. En este capítulo describimos el desarrollo del sistema. Se explican las distintas estructuras de datos utilizadas, los algoritmos realizados, decisiones de diseño tomadas, interfase de usuario, etc.
- CAP 5 Testeo
- CAP 6 Dificultades. En este capítulo tratamos algunos problemas que tuvimos y que retrasaron fuertemente la marcha del proyecto por lo que consideramos importante mencionarlos.
- CAP 7 Resultados y conclusiones.
- CAP 8 Referencias.
- CAP 9 Apéndices.

## 2 Definición de problemas

El problema clásico de Ruteo de Vehículos (VRP) se define como: “Determinar un conjunto de rutas de costo mínimo para una flota homogénea de vehículos que sirve a un conjunto de clientes dispersos geográficamente”. Modificando las condiciones o restricciones de dicho problema se pueden generar infinitas variantes. Para este proyecto elegimos una lista de problemas que consideramos abarcan todos los casos principales y más utilizados en la práctica y pueden ser ampliados para satisfacer las necesidades puntuales planteadas por un problema determinado. De esta forma cumplimos con el objetivo del proyecto en cuanto a generar un software que solucione la mayor cantidad de problemas de Ruteo de Vehículos y que sirva como base para el desarrollo de soluciones a problemas puntuales.

A continuación se detalla una lista con los problemas a tratar viendo su descripción [b].

### TSP

El TSP (Traveling Salesman Problem por sus siglas en inglés) se refiere al caso en que tenemos un solo depósito, un solo vehículo,  $N$  nodos que requieren servicio ubicados en una red (que puede ser dirigida o no), sin restricciones de capacidad del vehículo, sin restricciones de demanda de los clientes y sin restricciones de tiempo.

El objetivo es minimizar la distancia recorrida pasando por todos los nodos exactamente una vez volviendo al nodo de origen (depósito). Podemos definir la red como  $G = [N, A, C]$  siendo  $N$  el conjunto de nodos,  $A$  el conjunto de arcos y  $C$  la matriz de costos que determina la distancia de un nodo a otro. Si  $C_{ij} = d_{ij}$  entonces la distancia que hay que recorrer para ir del nodo  $i$  al nodo  $j$  es  $d_{ij}$ . Lo que se debe hallar es el Ciclo Hamiltoniano de costo mínimo. Si los costos son simétricos, o sea, si el costo de viajar de un lugar (nodo) a otro no depende de la dirección en que se viaje, estamos frente a un TSP simétrico; en caso contrario el TSP se llama asimétrico o dirigido.

Un claro ejemplo de aplicación de este problema es el recorrido que debe hacer un visitador médico relevando los pedidos.

#### Formulación Matemática.

Consideramos  $c_{ij} = c_{ji}$  = costo de ir del nodo  $i$  al nodo  $j$  y viceversa.

$$x_{ij} = \begin{cases} 1 & \text{si el arco } i-j \text{ se encuentra en la ruta obtenida como solución final} \\ 0 & \text{en otro caso} \end{cases}$$

Problema:

$$\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij}$$

$$\text{Sujeto a } \sum_{i=1}^n x_{ij} = b_j = 1 \quad (j = 1..n) \quad \text{Asegura que a cada nodo llega una sola arista.}$$

$$\sum_{j=1}^n x_{ij} = a_i = 1 \quad (i = 1..n) \quad \text{Asegura que de cada nodo sale una sola arista.}$$

$$x_{ij} = 0 \text{ o } 1 \quad (i,j = 1..n)$$

Estas ecuaciones no son suficientes para definir completamente el problema dado que no impiden la formación de sub-tours (ver Fig 1). El objetivo del problema es minimizar la distancia recorrida satisfaciendo la demanda de los nodos con un ciclo único. Con las 3 restricciones vistas anteriormente se puede presentar el siguiente caso:

$N = \{1, 2, 3, 4, 5, 6\}$  conjunto de nodos

$A = \{x_{12}, x_{23}, x_{31}, x_{45}, x_{56}, x_{64}\}$

En este caso tenemos 2 sub-tours o ciclos, formados uno por los nodos 1, 2 y 3 y el otro formado por los nodos 4, 5 y 6.

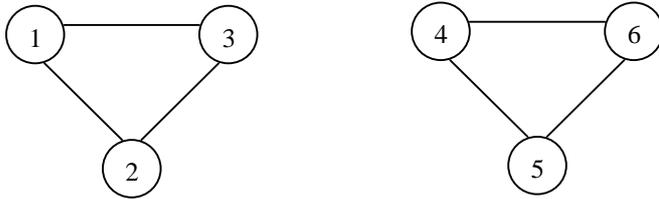


Fig 1

Para evitar esto debemos agregar la siguiente restricción al problema:

$$\sum_{i \in Q} \sum_{j \notin Q} x_{ij} \geq 1 \text{ para todo } Q \text{ subconjunto no vacío de } N$$

Esta ecuación nos asegura que de todo punto que pertenezca a un subconjunto de N sale una arista que termina en un nodo que no pertenece a dicho subconjunto. De esta forma aseguramos que existe un camino para ir de cualquier nodo a todos los otros, asegurando de esta forma que la solución hallada es un ciclo único.

**MTSP**

El “multi traveling salesman problem” (MTSP) es una generalización del TSP (un solo vehículo). La diferencia con el TSP es que se tiene una flota de M vehículos y un depósito. M vehículos deben visitar N nodos buscando minimizar la distancia total recorrida por los M vehículos. Cada vehículo debe visitar un sub-ciclo de nodos que comience y termine en el depósito común, y cada nodo debe ser visitado exactamente una vez por un vehículo.

**Formulación Matemática.**

Problema:

Minimizar  $\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$  {

Sujeto a  $\sum_{i=1}^n x_{ij} = b_j = \begin{cases} 1 & \text{si } j = 2,3, \dots, n \\ M & \text{si } j = 1 \end{cases}$

Asegura que a cada cliente llega una sola arista y que al depósito llegan M aristas

$$\sum_{j=1}^n x_{ij} = a_i = \begin{cases} 1 & \text{si } i = 2,3, \dots, n \\ M & \text{si } i = 1 \end{cases}$$

Asegura que de cada cliente sale una sola arista y que del depósito salen M aristas.

$$x_{ij} = 0 \text{ o } 1 \quad (i,j = 1..n)$$

Esta formulación matemática es la adaptación de la formulación matemática del TSP al MTSP

**VRP Clásico**

El VRP clásico es una generalización del MTSP (varios vehículos), lo cual lo hace aplicable a mayor cantidad de casos de la realidad. Las diferencias con el MTSP consisten en que en este caso los clientes tienen requerimientos de servicio que deben ser cumplidos (demanda), la flota de vehículos es limitada y los vehículos tienen capacidad limitada. Cada vehículo debe

visitar un sub-ciclo de nodos que comience y termine en el depósito común, y cada nodo debe ser visitado exactamente una vez por un vehículo.

Un ejemplo de este problema es el reparto de productos que debe hacer una empresa a sus clientes distribuidos geográficamente. Cada cliente tiene una determinada demanda y los vehículos deben comenzar y finalizar el día en el depósito.

### Formulación Matemática.

Podemos basarnos en la formulación matemática del MTSP pero faltan varias restricciones. Aquí no aparecen restricciones de cumplimiento de la demanda de los clientes, no exceder la capacidad de los vehículos, etc.

La formulación matemática completa del VRP clásico es la siguiente:

$$\text{Minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n \sum_{v=1}^{NV} c_{ij} x_{ij}^v \quad (\text{a})$$

$$\text{Sujeto a} \quad \sum_{i=1}^n \sum_{v=1}^{NV} x_{ij}^v = 1 \quad (j = 2, \dots, n) \quad (\text{b})$$

$$\sum_{j=1}^n \sum_{v=1}^{NV} x_{ij}^v = 1 \quad (i = 2, \dots, n) \quad (\text{c})$$

$$\sum_{i=1}^n x_{ip}^v - \sum_{j=1}^n x_{pj}^v = 0 \quad (v = 1, \dots, NV; p = 1, \dots, n) \quad (\text{d})$$

$$\sum_{i=1}^n d_i \left( \sum_{j=1}^n x_{ij}^v \right) \leq K_v \quad (v = 1, \dots, NV) \quad (\text{e})$$

$$\sum_{i=1}^n t_i^v \sum_{j=1}^n x_{ij}^v + \sum_{i=1}^n \sum_{j=1}^n t_{ij}^v x_{ij}^v \leq T_v \quad (v = 1, \dots, NV) \quad (\text{f})$$

$$\sum_{j=2}^n x_{1j}^v \leq 1 \quad (v = 1, \dots, NV) \quad (\text{g})$$

$$\sum_{i=2}^n x_{i1}^v \leq 1 \quad (v = 1, \dots, NV) \quad (\text{h})$$

$$x_{ij}^v = 0 \text{ or } 1 \quad \text{para todo } i, j, v \quad (\text{i})$$

Donde  $n$  = número de nodos,  $NV$  = número de vehículos,  $K_v$  = capacidad del vehículo  $v$ ,  $T_v$  = máximo tiempo posible para la ruta del vehículo  $v$ ,  $d_i$  = demanda del nodo  $i$  ( $d_1 = 0$ ),  $t_i^v$  = tiempo de servicio requerido por el vehículo  $v$  en el nodo  $i$  ( $t_1^v = 0$ ),  $t_{ij}^v$  = tiempo de viaje del vehículo  $v$  del nodo  $i$  al nodo  $j$ ,  $c_{ij}$  = costo del viaje del nodo  $i$  al nodo  $j$ ,  $x_{ij}^v = 1$  si el arco  $i$ - $j$  es atravesado por el vehículo  $v$  y 0 en otro caso.

La ecuación (a) indica que el objetivo es minimizar la distancia total recorrida. También se pueden minimizar costos sustituyendo  $c_{ij}$  por un coeficiente  $c_{ij}^v$  que depende del vehículo. Las ecuaciones (b) y (c) que la demanda de cada nodo es cumplida exactamente por un vehículo. La continuidad de la ruta es asegurada por la ecuación (d), si un vehículo entra a un nodo con demanda, debe salir de dicho nodo. La ecuación (e) establece la restricción de capacidad de los vehículos y la

ecuación (f) las restricciones de tiempo total que demora en recorrerse una ruta. Las ecuaciones (g) y (h) aseguran que la disponibilidad de vehículos no es superada.

## Extensiones del problema clásico

Al VRP clásico se le pueden aplicar diferentes extensiones o variantes. En este caso consideraremos los problemas con más de un depósito, con flota heterogénea y con ventanas de tiempo.

### MDVRP

Este tipo de problema es similar al VRP (ruteo de  $N$  vehículos con un depósito único y clientes con demanda preestablecida y sin ventanas de tiempo) con la diferencia de que se tiene un conjunto de depósitos y no un depósito único como ocurría en el caso anterior. De esta forma tenemos una flota de  $M$  vehículos que salen de  $D$  depósitos y deben satisfacer la demanda de  $N$  nodos. Los vehículos deben salir de un depósito y volver al mismo.

Los ejemplos prácticos de aplicación son similares a los del VRP pero para empresas más grandes que tengan varios depósitos. Un ejemplo sería un supermercado que en una ciudad tenga varias sucursales y recepcione (por internet por ej.) pedidos de sus clientes. Dada la demanda y ubicación geográfica de cada cliente se formarán las rutas a recorrer por cada vehículo de reparto de mercadería que saldrán de las diferentes sucursales del supermercado.

#### Formulación Matemática.

Pequeñas variaciones a la formulación matemática (ecuaciones (a) a la (i)) vista para el VRP clásico permiten establecer la formulación matemática para el MDVRP. Consideramos los nodos  $1, 2, \dots, M$  como depósitos, obtenemos las nuevas ecuaciones modificando el índice en las restricciones (b) y (c) a  $(j = M+1, \dots, n)$  y  $(i = M+1, \dots, n)$  respectivamente, y modificando las restricciones (g) y (h) por las siguientes:

$$\sum_{i=1}^M \sum_{j=M+1}^n x_{ij}^v \leq 1 \quad (v = 1, \dots, NV) \quad (g')$$

$$\sum_{p=1}^M \sum_{i=M+1}^n x_{ip}^v \leq 1 \quad (v = 1, \dots, NV) \quad (h')$$

### Problemas con Flota Heterogénea

Ahora consideraremos una situación en que el distribuidor decide, por conveniencia y confiabilidad, alquilar, en lugar de comprar, vehículos por un cierto tiempo. El número de vehículos de cada tipo necesario para satisfacer la demanda en forma efectiva debe ser determinado con una política de ruteo coherente para esos vehículos. Asumimos que cada tipo de vehículo tiene un costo de alquiler fijo y un costo de ruteo variable que es proporcional a la distancia recorrida. Cada tipo de vehículo tiene su capacidad y su costo fijo. El componente del costo variable depende del combustible, mantenimiento y mano de obra.

Para resolver este tipo de problemas utilizaremos el algoritmo de Clarke y Write (CW) tradicional [b], algoritmo de los ahorros modificando la función utilizada para calcular los ahorros. En este algoritmo dos rutas conteniendo clientes  $i$  y  $j$  pueden ser unidas si: a)  $i$  y  $j$  pertenecen a rutas diferentes, b)  $i$  y  $j$  no son puntos interiores a las rutas, c) la demanda de la combinación de ambas rutas no excede la capacidad de los vehículos.

Recordamos que la decisión sobre cuáles rutas unir se basa en el cálculo del ahorro que dicha unión permite obtener. Los ahorros se calculan como  $s(i, j) = c(0, i) + c(0, j) - c(i, j)$ , siendo  $c(i, j)$  el costo de ir desde el punto  $i$  al punto  $j$  y siendo  $0$  un depósito. El problema que plantea este método para el caso de flota heterogénea es que no considera los costos fijos de los vehículos. Debemos ampliar el concepto de ahorro incluyendo ahora dichos costos además de los costos de ruteo. A este método se le llama método "Combined Savings" (CS) [b].

Sea  $F$  una función en los reales que dada  $z$  (demanda de una ruta) devuelve el costo fijo del menor vehículo capaz de satisfacer la demanda  $z$ . Consideremos una ruta  $I$  que tiene a  $i$  como nodo "límite" (nodos conectados con el depósito) y una ruta  $J$  que tiene a  $j$  como nodo "límite". Si la demanda total de estas rutas es  $z_i$  y  $z_j$  el ahorro obtenido al unir dichas rutas es:  $s'(i, j) = s(i, j) + F(z_i) + F(z_j) - F(z_i + z_j)$ . Este método tiene el inconveniente que los ahorros calculados son inmediatos.

Por ejemplo supongamos que tenemos las rutas  $I$ ,  $J$  y  $K$  con una demanda de 100 cada una servidas por vehículos de capacidad 100. Supongamos también que el próximo vehículo más grande tiene una capacidad de 300 y su costo fijo es considerablemente mayor al de los vehículos de capacidad 100. Puede ocurrir que la combinación de las tres rutas siendo

servidas por un vehículo de capacidad 300 sea más barata que las tres rutas por separado. Pero el algoritmo CS encontrará dicha solución? No necesariamente. Dado que las rutas se combinan de a dos es probable que al combinar las rutas I y J por ej. el ahorro sea negativo ya que se requiere un vehículo de capacidad 300 el cual tiene un costo fijo elevado por lo que dicha combinación sería descartada. En este caso CS no considera que el vehículo de capacidad 300 tiene capacidad de 100 unidades inutilizada, la cual puede ser útil a la hora de unir una nueva ruta sin aumentar los costos fijos.

Esto genera una variante de este algoritmo, llamado Opportunity Savings Algorithms [b]. A los ahorros vistos en el punto anterior se agregan los “Opportunity savings” (OS), los cuales son una función de la capacidad no utilizada del vehículo que cubre la nueva ruta formada.

Un tipo de Opportunity Saving Algorithm es el Optimistic Opportunity Savings (OOS). En este caso los OS se definen como el costo fijo del menor vehículo que puede servir la capacidad inutilizada del nuevo vehículo. Es decir que si tenemos vehículos de capacidades 50, 100 y 200 y formamos una nueva ruta con demanda 120 servida por un vehículo con capacidad 200 el ahorro  $s'' = s' + OS$  siendo OS el costo fijo del vehículo de capacidad 100 por ser el menor vehículo capaz de cubrir las 80 unidades de capacidad inutilizada. La explicación a esta fórmula está dada por el hecho de que se asume que se encontrará una ruta de demanda 80 la cual se agregará a la ruta recién formada ahorrándose el costo del vehículo de 100 que servía a la ruta de 80. Sea  $P(z)$  la capacidad del menor vehículo capaz de satisfacer la demanda de la ruta  $z$  entonces:  $s''(i,j) = s(i,j) + F(z_i) + F(z_j) - F(z_i + z_j) + F(P(z_i + z_j) - z_i - z_j) = s'(i,j) + F(P(z_i + z_j) - z_i - z_j)$ .

Otro tipo de Opportunity Saving Algorithm es el Realistic Opportunity Savings (ROS). Sea  $F'(z)$  análoga a  $F(z)$  salvo que  $F'(z)$  representa los costos fijos del mayor vehículo cuya capacidad es menor o igual a  $z$ . Sean I y J dos rutas con demanda  $z_i$  y  $z_j$  y sean  $i$  y  $j$  sus puntos terminales. Asumimos que  $z_i \geq z_j$ .

Si  $F(z_i + z_j) = F(z_i)$  el ahorro es el mismo que el calculado en el caso del CS, es decir  $s'(i,j)$ . Si  $F(z_i + z_j) > F(z_i)$  debemos considerar los OS por lo que  $s'''(i,j) = s'(i,j) + F'(P(z_i + z_j) - z_i - z_j)$ . Datos empíricos demuestran que ROS es superior a OOS y CS dado que brinda mejores resultados en 7 de 12 muestreos obteniendo mejores tiempos en los 12 muestreos.

En resumen tenemos los siguientes métodos y sus respectivos cálculos de ahorros:

Algoritmo	Ahorro	Fórmula para calcular el ahorro
CW	$s(i,j)$	$c(0,i) + c(0,j) - c(i,j)$
CS	$s'(i,j)$	$s(i,j) + F(z_i) + F(z_j) - F(z_i + z_j)$
OOS	$s''(i,j)$	$s'(i,j) + F(P(z_i + z_j) - z_i - z_j)$
ROS	$s'''(i,j)$	$s'(i,j) + F'(P(z_i + z_j) - z_i - z_j)$

## Problemas con Ventanas de Tiempo

Los distintos puntos del sistema (ya sean depósitos, puntos intermedios o destinos) pueden tener restricciones de tiempo. El manejo más común de las restricciones de tiempo se realiza a través de las ventanas de tiempo.

Una ventana de tiempo de dos dimensiones  $[s,t]$  para una entidad determina que dicha entidad debe ser visitada en el intervalo de tiempo que va desde  $s$  hasta  $t$ . Si estamos hablando por ejemplo de un comercio de venta de ropa podemos establecer que la mercadería puede llegar al mismo entre las 10hs. y las 18hs., por lo que su ventana de tiempo será  $[10,18]$ .

También podemos expresar solamente un tiempo mínimo o máximo. Una ventana de tiempo de una dimensión es  $[-\infty,t]$  o  $[t,+\infty]$ . La primera ventana de tiempo implica que el servicio debe ser llevado a cabo antes del tiempo  $t$  y la segunda implica que el servicio debe ser llevado a cabo luego del tiempo  $t$ .

Una determinada entidad puede estar asociada a más de una ventana de tiempo. Por ejemplo, un comercio que está abierto de 8hs. a 12hs. y de 14hs. a 18hs. y solo en ese horario puede recibir la mercadería.

Las ventanas de tiempo pueden representar también días de la semana, por ejemplo se puede decir que una entidad está disponible sólo los martes, jueves y sábados. Una vez que los días de la semana son asignados, podría haber ventanas de tiempo y restricciones de precedencia entre tareas para las tareas que se realizarán durante esos días.

Algunas aplicaciones de los problemas con ventanas de tiempo son:

- Ruteo de ómnibuses escolares
- Ruteo de camiones con carrocería con cargas parciales o totales
- Ruteo de vendedores callejeros

## ***CPP***

El problema del cartero chino requiere la determinación de un ciclo de costo mínimo que pase por todos los arcos de la red al menos una vez (la demanda está en los arcos). Puede ser dirigido o no dirigido, según si los arcos son dirigidos o no. Las dos variaciones pueden resolverse por algoritmos con tiempo de ejecución polinómico.

El problema del cartero chino *mixto* tiene algunos arcos dirigidos y otros no, este problema es *NP*-duro.

En nuestro caso trataremos el problema del CPP no dirigido dado que los demás están fuera del alcance de este taller.

Algunas aplicaciones del problema del cartero chino son: ruteo de vendedores callejeros, carteros, inspectores de las líneas eléctricas, etc. Como en algunas situaciones la cantidad de clientes que requieren servicio es muy alta, identificarlos individualmente sería muy engorroso; en estos casos consideramos al problema del cartero chino como la variación continua del TSP discreto. Aquí un arco sustituye a una cantidad de clientes.

Los arcos tienen requerimientos de servicio que deben ser cumplidos, la flota de vehículos es limitada y los vehículos tienen capacidad limitada.

## ***DARP***

En problemas del tipo DARP (dial-a-ride-problem por sus siglas en inglés) los clientes llaman al “despachador” solicitando un servicio. Cada cliente especifica el punto de carga y entrega de lo que se vaya a transportar y quizás también el período de tiempo en que desea que sea realizada cada acción. Si todos los clientes reclaman servicio inmediato estamos frente a un DARP dinámico o DARP en tiempo real. Si todos los clientes llaman con anticipación de forma tal que la base de datos con los requerimientos de los clientes este completa antes de comenzar a hallar la solución estamos frente a un DARP estático.

En algunos casos el horario de carga (pickup) o entrega (delivery) es especificado con anticipación y la otra acción debe ser realizada dentro de un período de tiempo determinado desde el horario ya dado. Es decir, se fija uno de los extremos y el período máximo de tiempo que puede transcurrir entre la carga y entrega en lugar del caso tradicional en que se indican horario de carga y horario de entrega. En este caso estamos frente a un problema con ventanas de tiempo de dos dimensiones.

### Software comercial, VRP y extensiones

En la siguiente tabla podemos apreciar algunos paquetes de software difundidos comercialmente y que casos resuelve cada uno de ellos (ver Apéndice 1).

	TSP	TSP TW	TSP FH	VRP	VRP TW	VRP FH	MDVRP	CPP	DARP
DynaRoute	-	-	-	-	-	-	-	-	Si
GeoRoute 5	Si	Si	Si	Si	Si	Si	Si	-	-
GeoPostal	-	-	-	-	-	-	-	Si	-
RouteSmart	Si	Si	Si	Si	Si	Si	Si	-	-
SHIPCONS II	Si	Si	Si	Si	Si	Si	Si	-	-
TransCad	Si	Si	Si	Si	Si	Si	Si	Si	-
Trapeze-Taxi	-	-	-	-	-	-	-	-	Si
Direct Route	Si	Si	Si	Si	Si	Si	S/i	-	-
GeoRoute-Municipal	-	-	-	-	-	-	-	Si	-
LoadExpress Plus	Si	Si	Si	Si	Si	Si	Si	-	-
RiMMS	Si	s/i	s/i	Si	s/i	s/i	S/i	-	-
RoutePro Dispatcher	Si	Si	Si	Si	Si	Si	Si	-	-
Routronics 2000	-	-	-	-	-	-	-	-	Si
STARS	Si	Si	Si	Si	Si	Si	Si	-	-
Optrak	Si	s/i	s/i	Si	s/i	s/i	Si	-	-
TruckStops for Windows	Si	Si	Si	Si	Si	Si	Si	-	-
ArcLogistics Route	Si	Si	Si	Si	Si	Si	Si	-	-

Notar que no existe ningún paquete comercial que resuelva todos los casos.

Podemos observar que existen tres grupos de paquetes comerciales:

- Los que resuelven el problema del ruteo de vehículos con muchos depósitos, flotas heterogéneas y time-windows (donde el TSP, TSPTW, TSPFH, VRP, VRPTW Y VRPFH son casos particulares)
- Los que resuelven el problema del CPP
- Los que resuelven el problema DARP

Vemos entonces que hay 3 tipos de paquetes claramente diferenciados y que no son comunes los paquetes que resuelvan determinados problemas de un tipo y problemas de otros tipos. Ver que ninguno de los que resuelve DARP resuelve otro tipo de problemas, ninguno de los que resuelve CPP resuelve otro tipo de problemas y ninguno de los que resuelve VRP y sus extensiones resuelven DARP o CPP.

### 3 Requerimientos de Usuario

Los requerimientos de usuario son relevados ante el Ing. José Comas integrante del área de Optimización de la empresa I.C.A. S.A.

- Generar un software genérico que resuelva la mayor cantidad posible de problemas de ruteo, logrando de esta forma un paquete genérico.
  - Deberá servir de base para futuros proyectos de casos más específicos.
  - Los problemas a resolver son:
    - TSP
    - MTSP
    - VRP
    - MDVRP
    - VRP con ventanas de tiempo (VRPTW)
    - MDVRP con ventanas de tiempo (MDVRPTW)
    - VRP para flota heterogénea
    - CPP
    - DARP
- Se selecciona esta lista de problemas pues cubren los problemas clásicos y sus principales extensiones.
- Interfase gráfica para la entrada y salida de datos. La interfase gráfica debe permitir ingresar los datos seleccionando los archivos a utilizar, visualizar gráficamente los datos ingresados y visualizar la solución obtenida.
  - Los datos geográficos serán representados utilizando la herramienta MapObjects brindada por el usuario.
  - La interfase gráfica se desarrollará en Visual Basic dado que permite óptimos resultados visuales y gran interacción con MapObjects.
  - Los datos geográficos se entregarán en formato shapfile. Dichos formato es nativo de la firma ESRI y consta de tres archivos: un archivo .shp con los datos geográficos, un archivo .dbf con la información de las entidades (clientes, depósitos, rutas, etc.) y un archivo índice que realiza la conexión entre los dos mencionados anteriormente.
  - Los datos de la flota de vehículos se ingresarán directamente a través de la interfase gráfica.
  - El motor de la base de datos y la resolución de los distintos algoritmos se hará en C++.
  - Las estructuras de datos a utilizar quedan a elección de los desarrolladores.
  - Como resultado se espera un archivo .txt con las rutas generadas. Cada ruta estará representada por un depósito y una lista de clientes. Deberán poder visualizarse los resultados en el mapa mostrado en la interfase gráfica.

## 4 Implementación

En esta sección vemos todo lo referente al desarrollo del sistema en sus diferentes partes. Se explican las distintas estructuras de datos utilizadas, los algoritmos realizados, decisiones de diseño tomadas, interfase de usuario, etc.

### *Arquitectura del Sistema y Lenguajes de Programación*

Los lenguajes de programación utilizados fueron elegidos siguiendo las necesidades del usuario y respetando sus preferencias en cuanto a qué lenguajes utilizar.

Para el desarrollo de la interfase gráfica con el usuario utilizaremos la herramienta MapObjects y realizaremos programación en Visual Basic. Esto nos permite una efectiva y cómoda visualización de datos geográficos facilitando de esta manera el trabajo de los usuarios y permitiendo una buena comunicación con el sistema en general. De esta forma se podrá recibir información sobre los clientes, depósitos y rutas que forman el sistema sobre el cual se va a trabajar. Los datos se recibirán como archivos shp siendo luego transformados a un formato manejable por parte del sistema.

El motor de la base de datos y la programación de los algoritmos de resolución de los diferentes casos de ruteo se hará en C++ dado que nos permite una mayor rapidez que otros lenguajes similares.

### *Elección de Algoritmos*

En la realización de los algoritmos se prioriza su reusabilidad y rapidez frente a la utilización de memoria. Se busca hacer un paquete genérico que luego pueda ser mejorado y adaptado a las diferentes necesidades de los posibles clientes por lo que es primordial con los algoritmos sean reusables y no sean lentos por motivos de su diseño básico. El tema de la utilización de memoria si bien debe ser tenido en cuenta no es considerado primordial de acuerdo a la solicitud del usuario.

Hemos elegido el algoritmo de los ahorros de Clark y Wright [b] para resolver los distintos tipos de problemas de ruteo.

El algoritmo, en el caso general, consta de los siguientes pasos:

- 1) Dado un depósito y  $n$  clientes que serán servidos por ese depósito, crear un ciclo entre el depósito y el cliente para cada cliente.
- 2) Calcular los ahorros, esto es cuanto ahorramos al ir del depósito al cliente A, luego al cliente B y luego al depósito, en lugar de ir y volver a cada cliente desde el depósito.  
Formalizando,  $s_{ij} = c_{0i} + c_{0j} - c_{ij}$ , siendo  $c_{ij}$  la distancia (o el costo) entre los nodos  $i$  y  $j$ . El nodo 0 representa el depósito.
- 3) Ordenar los ahorros de mayor a menor.
- 4) Comenzando por el ahorro mayor, unir los nodos  $i$  y  $j$  correspondientes y quitar los arcos  $i-1$  y  $1-j$ . Repetir este paso hasta llegar a la solución.

Existen otros algoritmos entre los cuales podemos destacar los procedimientos de inserción, que se basan en iniciar el procedimiento solo con el depósito e ir insertando nodos para lo cual existen diferentes criterios (por ej.: inserción del más cercano, inserción más barata, etc.).

Elegimos el algoritmo de Clark y Wright debido a que se adapta mas facilmente a los distintos problemas que vamos a resolver. Por ejemplo, el algoritmo para solucionar el TSP es un caso particular del algoritmo para solucionar el VRP, que a su vez son casos particulares del MDVRP.

Dado que lo que buscamos en todos los casos es ahorrar la distancia recorrida vamos a necesitar reiteradamente hallar el camino mas corto entre 2 nodos o esquinas. El algoritmo utilizado para calcular los caminos más cortos entre cada par de clientes es Dijkstra. Esto se debe a que es un algoritmo que utilizado en forma correcta (o sea sin utilizar mucha memoria) puede ser igualmente eficiente.

Los resultados obtenidos al ejecutar Dijkstra son utilizados reiteradas veces por los diferentes algoritmos desarrollados, por lo que surge la idea de realizar un Dijkstra completo (de todos los nodos) al comienzo y almacenar los resultados.

Se evalúan dos posibilidades a la hora de guardar los resultados obtenidos al ejecutar Dijkstra. Las posibilidades son las siguientes: a) guardar los datos en memoria, b) guardar los datos en un archivo y acceder a ellos cada vez que sean necesarios. Para problemas con pocos nodos (menos de 100) es claro que la opción a) es la mejor pero debemos tener en cuenta que al realizar un software genérico, el mismo debe funcionar correctamente también para problemas de gran tamaño. El almacenamiento en memoria nos brinda el más rápido acceso a los datos y evita calcular Dijkstra muchas veces pero tiene un altísimo costo en cuanto a utilización de la memoria, pues para representar una ciudad de 15.000 esquinas serían

necesarios alrededor de 225 Mb. Si bien el ahorro de memoria no es una de nuestras prioridades si debemos realizar algoritmos con requerimientos adecuados a los estándares actuales por lo cual esta opción fue descartada.

El almacenamiento en disco duro nos soluciona el problema de la utilización de mucha memoria pero se genera un archivo que es muy grande lo cual no es conveniente. El acceso a dicho archivo resulta lento y debe hacerse cada vez que se requiera un dato por lo que esta opción también es descartada.

Surgen entonces nuevamente dos posibilidades: a) almacenar algunos datos (los mas utilizados) en memoria y otros en disco, b) calcular Dijkstra cada vez que sea necesario buscando siempre reducir la cantidad de veces que se ejecuta dicho algoritmo. La opción a) es descartada pues no siempre es fácil o probable determinar cuáles son los datos más utilizados lo que imposibilita implementar correctamente el almacenamiento de datos en memoria y disco duro. Optamos por la opción b) ya que no nos presenta ningún problema en cuanto a utilización de la memoria y el tiempo utilizado al ejecutar Dijkstra es bajo.

Realizamos dos versiones del algoritmo de Dijkstra. Uno que nos da el costo y camino para ir de un nodo al resto, y otro que nos da el costo, camino y tiempo necesario para ir de un nodo al resto. El primero se utilizará para problemas sin restricciones de tiempo y el segundo para el resto.

Se evalúa la posibilidad de realizar versiones reducidas del algoritmo buscando reducir el tiempo de ejecución para casos puntuales. Dichas versiones son: a) Algoritmo que devuelve solo los costos y no los caminos b) Algoritmo que devuelve el costo y camino para ir de un nodo a otro en particular y no de un nodo a todos los otros como en el caso habitual. Finalmente estas versiones reducidas no se realizaron pues la mejora obtenida no era significativa. En el caso a) el algoritmo es exactamente el mismo pues para calcular los costos se deben calcular los caminos. En el caso b) se podría obtener cierta mejora al llegar al mejor camino hacia el nodo buscado pero dicha mejora no es significativa y además no son muchos los casos en que se desea el camino más corto de un nodo a otro en particular, en general se busca el camino mas corto de un nodo a un conjunto determinado de nodos.

## ***Interfase de usuario***

La interfase del usuario es la parte del software implementado con la que el usuario interactuará. En esta sección describimos:

- Generalidades: Descripción de la interfase, funcionalidad brindada
- Herramientas utilizadas
- Arquitectura de la comunicación entre la interfase y el motor de cálculo
- implementación

### **I. Generalidades**

El objetivo de la interfase (y del software en general), es que el usuario seleccione archivos del tipo shapefile (formato de archivos que representan información geográfica, propiedad del ESRI) que son usados para representar las calles de una ciudad, clientes y depósitos ubicados en la misma, y que sobre esta información pueda resolver ciertos problemas predefinidos (TSP, VRP, MDVRP, MDVRPTW, DARP). Nuestro objetivo es que la forma de elección de estos shapefile, así como su visualización y la visualización del resultado, se hagan en forma similar a como lo hace ArcView (software del ESRI que permite visualizar, construir y consultar shapefiles), debido a que es probable que el usuario que maneje shapefiles también use dicha herramienta.

### **II. Herramientas utilizadas**

Debido a la característica netamente gráfica de la interfase a construir, elegimos el lenguaje de programación Visual Basic (de Microsoft) para implementarla. Además, contamos con la herramienta MapObjects (MapObjects es un conjunto de componentes de software de mapeo creado por el ESRI que permite manipular fácilmente shapefiles, brindando funciones tales como visualizar un shapefile, hacer zoom sobre un mapa, dibujar diversos objetos como líneas, puntos y círculos, seleccionar información mediante consultas SQL, entre otras), que se integra a Visual Basic.

### **III. Arquitectura de la comunicación entre la interfase y el motor de cálculo.**

Debido a que el motor de cálculo es programado en forma independiente de la interfase de usuario (y en otro lenguaje de programación, C++), es necesario implementar algún mecanismo de comunicación entre estos, de modo que el motor de cálculo tenga acceso a los datos ingresados por el usuario. Para esto consideramos dos alternativas: pasarle como parámetros al motor de cálculo los nombres y la ubicación de los shapefiles seleccionados por el usuario y que dicho motor se encargara

del procesamiento de los mismos, o realizar el procesamiento de los datos de los shapefiles en Visual Basic y pasarle los datos obtenidos al motor de cálculo. Examinaremos a continuación ambas alternativas:

### **Procesamiento dentro del motor de cálculo**

Esta alternativa tiene varias desventajas, por lo que fue rápidamente desechada. La desventaja mas importante es que no contamos con una herramienta, como lo es MapObjects en Visual Basic, que nos permita una fácil manipulación de los shapefiles con el lenguaje C. Es decir, si hacemos el procesamiento desde C, debemos leer los shapefiles teniendo en cuenta su formato (que es complejo) y en modo binario (Ver ESRI Shapefile Technical Description para ver el formato de los shapefile). Esto tiene la desventaja, en primer lugar, de hacer que el código sea poco claro (e innecesariamente largo), y además genera una dependencia muy fuerte con el formato propuesto por ESRI (si el formato cambiara, se deberían modificar los programas, en cambio si se usara la herramienta MapObjects sólo habría que hacer un upgrade de la misma y la recompilación del código).

Cabe aclarar que el hecho de no contar con una herramienta como MapObjects para C no es la única desventaja que encontramos para desechar esta alternativa, ya que existe una versión de MapObjects para C++ (aunque no sabemos si es compatible con la versión del compilador usado por nosotros), y además poseemos un utilitario dado por el Ing José Comas que permite traducir un shapefile a un archivo de texto (lo que nos permite trabajar con shapefiles con relativa comodidad a pesar de no poseer una herramienta adecuada). La otra desventaja grande que surge con esta alternativa es que, aunque leyéramos aquí el shapefile, es imprescindible leerlo también desde el módulo de la interfase para dibujar el mapa y luego la ruta. Es decir, estaríamos leyendo el shapefile dos veces, lo que implica un gasto de tiempo que puede ser considerable o no dependiendo del tamaño del mapa (la cantidad de los clientes y los depósitos es despreciable en comparación con el tamaño del mapa). Para fijar ideas, el mapa de Montevideo ocupa unos tres megabytes en disco, por lo que leerlo una vez o dos veces genera diferencias de tiempo importantes.

Por último, el procesamiento de los shapefiles dentro del motor de cálculo hace casi imposible la proyección de los clientes sobre las rutas, debido a que estas (las rutas) se representan solo como un par de puntos, y no hay un forma eficiente de encontrar, para un punto (el cliente), cual es el par de puntos (la ruta) que forman la recta mas cercana a el, para luego realizar la proyección. Un detalle de por qué y como fue realizada la proyección puede encontrarse en el punto IV (implementación).

### **Procesamiento desde Visual Basic**

La otra alternativa, casi seleccionada por descarte de la alternativa anterior debido a la cantidad de desventajas que presenta, fue la de procesar los datos geográficos desde Visual Basic, para luego pasarlos al motor de cálculo. Cabe señalar que todas las desventajas que presenta la alternativa anterior se resuelven con esta nueva alternativa: es fácil manejar los shapefiles usando MapObjects y es fácil realizar la proyección de los clientes usando las funciones que brinda MapObjects.

Lo que resta por definir es la forma en que los datos geográficos son pasados desde Visual Basic al motor de cálculo, surgiendo una vez mas dos alternativas posibles: pasar los datos geográficos como parámetros al motor de cálculo o escribir archivos planos de texto con los datos y que el motor de cálculo los lea.

La primera alternativa tiene como ventaja, en comparación con la segunda, la performance (ya que es mucho mas barato en términos de tiempo crear una lista en memoria y pasarla como parámetro que crear un archivo de texto y luego leerlo). El problema aquí es que estamos ocupando memoria RAM, que puede llegar a ser un recurso escaso (no queremos que el software desarrollado tenga grandes requerimientos de memoria), debido al gran uso de memoria que hace el motor de cálculo. Considerando que, por una cuestión de performance, los datos geográficos van a estar permanentemente en la RAM durante la ejecución del sistema (son necesarios para dibujar las rutas, y seria poco eficiente que cada vez que haya que dibujar una ruta haya que leer el shapefile), decidimos desechar la primera alternativa y elegir la creación de archivos de texto con campos separados por comas para ser leídos por el motor. Notar que la filosofía subyacente durante la creación de la interfase fue priorizar tanto la utilización de la memoria como la performance.

En total se usan cinco archivos de texto para la comunicación entre la interfase gráfica y el motor de cálculo, que detallamos a continuación:

1) Archivo CLIENTS.TXT. En este archivo, que es creado por la interfase y leído por el motor de cálculo, se guarda la información de los clientes. Su formato varía dependiendo del tipo de algoritmo que se vaya a correr (DARP u otro). El formato para el algoritmo DARP es el siguiente:

operación, id del cliente en el dbf, id interno del cliente, id del nodo a la izq, id del nodo a la der., distancia al nodo de la izq, distancia al nodo de la derecha, demora para ir al nodo de la izq., demora para ir al nodo de la der., hora mínima de atención, hora máxima de atención, tiempo de atención (en minutos)

donde:

- tipo de operación es 1 = pick up, 0 = delivery, 2 = deposito
- las time windows están expresadas en minutos (por ej. 15h30' =  $60 \times 15 + 30 = 930$ )

Para el caso de un registro con operación = 2 (significa que en vez de un cliente es un deposito), los campos hora mínima y máxima de atención, así como el campo tiempo de atención están en 0.

El formato de CLIENTS.TXT para otros algoritmos es el siguiente:

tipo de registro, id interno del cliente, id del nodo de la izquierda, id del nodo de la derecha, distancia al nodo de la izquierda, distancia al nodo de la derecha, demora al nodo de la izquierda, demora al nodo de la derecha, demanda, hora mínima de atención, hora máxima de atención, tiempo de atención

donde:

- tipo de registro = "C" (cliente) o "D" (deposito)
- las time windows están expresadas en minutos

En el caso de un registro de tipo "D" (deposito), la hora mínima y máxima de atención, el tiempo de atención y la demanda están en 0

Una aclaración importante es que cuando nos referimos a nodo de la izquierda y de la derecha, este no necesariamente tiene que ser un nodo de tipo esquina, sino que se contempla la posibilidad que en una misma cuadra hayan varios clientes.

2) Archivo INFO.TXT. Este archivo, creado por la interfase y leído por el motor de cálculo, contiene información estadística que será usada por el motor para la creación de sus estructuras dinámicas de datos.

La estructura de este archivo es la siguiente:

cantidad de calles  
 cantidad de clientes  
 cantidad de depósitos

3) Archivo STREETS.TXT. Este archivo, creado por la interfase y leído por el motor de cálculo, contiene la información relativa a las rutas que componen el mapa con el cual se está trabajando. Su formato es el siguiente:

nodo x, nodo y, distancia(x,y), tiempo de demora para ir de x a y, indicación si es flechada

donde :

- nodo x e y son ids internos de los nodos que determinan la calle
- distancia(x,y) es la distancia euclidiana, no esta medida en ninguna unidad en particular (pues se toman como referencias las coordenadas de los puntos), y su uso es interno.
- indicación si es flechada: "S" = Si, "N" = No

4) Archivo VEHICLES.TXT. Este archivo, creado por la interfase y leído por el motor de cálculo, contiene la información de la flota de vehículos que el usuario ingresó. El formato del archivo varía si la flota a usar es homogénea (todos los vehículos tienen la misma capacidad) o heterogénea (vehículos con distinta capacidad).

Para el caso de flota homogénea, el formato es el siguiente:

"O"  
 capacidad

donde capacidad es la capacidad de todos los vehículos.

Para una flota heterogénea, el formato es el siguiente:

"E"  
 cap\_1, CF\_1, cant\_1  
 cap\_2, CF\_2, cant\_2  
 :  
 :  
 cap\_n, Cf\_n, cant\_n

donde:

- cap es la capacidad del vehículo
- CF es el costo fijo
- cant es la cantidad de vehículos de capacidad cap.

5) Archivo RUTAS.TXT. Este archivo, generado por el motor de cálculo y leído por la interfase gráfica, es una sucesión de identificadores de nodos que forman una o mas rutas. Su formato es:

```
Id 1-1
Id 1-2
:
:
Id 1-N
0
Id 2-1
Id 2-2
:
:
Id 2-n
```

En el ejemplo vemos un archivo que representa dos rutas, cada ruta se separa por un identificador 0.

#### IV. Implementación

En esta sección describimos la forma en que se realizan las proyecciones de los clientes sobre las rutas y las estructuras de datos utilizadas en la interfase. El resto de la implementación puede ser comprendida fácilmente mirando el código.

##### Proyección de los clientes sobre las rutas.

Dado que los clientes son ingresados por parte del usuario sobre el mapa (con herramientas como por ejemplo ArcView), es probable que el punto que representa al cliente no quede sobre la recta que representa la calle. Es muy importante que el cliente se represente justo sobre la calle, pues de no hacerlo, en las estructuras de datos del motor de cálculos este cliente queda aislado de la red vial. Para esto, la solución que implementamos es proyectar los puntos que el usuario ingresa como clientes (y depósitos) sobre las rectas que representan las calles del mapa, e internamente trabajar con estos puntos proyectados.

##### Estructuras de datos utilizadas.

El algoritmo de carga de datos de la interfase lee el shapefile y convierte las coordenadas x e y de cada punto a un identificador interno. Para mantener esto, necesitamos una estructura de datos que permita, por un lado establecer rápidamente si un punto representado por un par de coordenadas ya se encuentra en la lista (para no insertar un punto varias veces), y por otro lado dado un identificador, devolver eficientemente las coordenadas x e y que dicho punto representa.

Para esto tenemos dos alternativas: guardar esta información en una base de datos (en particular Access), que nos permitirá realizar ambas consultas mediante la utilización de índices, o crear una estructura dinámica en memoria RAM.

Construimos ambas soluciones y comparamos tiempos. Los resultados fueron los siguientes: para cargar un mapa de aproximadamente 1500 esquinas, la opción con base de datos insumió cerca de 3 minutos 20 segundos, mientras que la opción de usar memoria dinámica insumió unos 30 segundos. La desventaja que tendría el uso de memoria RAM sería justamente el hecho de ocupar ese recurso, cuando sabemos que el motor de cálculo lo va a usar en forma intensa, pero estudiando la estructura de datos que proponemos, vemos que esta ocupa  $20n$  bytes, con  $n$  = cantidad de esquinas + clientes + depósitos, por lo que para el mapa de referencia (1500 esquinas) tendríamos que la estructura de datos ocupa tan solo alrededor de 30 K de memoria. La brecha entre tiempo y espacio ocupado se agrandaría aun mas si consideramos un mapa mas grande, como por ejemplo el de la ciudad de Montevideo (alrededor de 30000 esquinas).

Por lo tanto parece claro que es conveniente tener los datos del mapa en memoria. Para ello construimos una lista doblemente indizada, que tiene como índices el identificador interno de los nodos por un lado y la pareja de coordenadas x e y por otro. Para ello usamos un array (donde el índice del array es usado como el identificador del cliente) y dentro de dicho array ordenamos los elementos por coordenadas x e y, valiéndonos de un campo que actúa como “puntero” a las distintas celdas del array (similar a la implementación de listas con cursores).

Usamos además una lista auxiliar de estructura similar a la anterior, pero que tiene como índice las coordenadas x, y y la distancia entre el punto y el nodo perteneciente a la calle de mas a la izquierda. Esta estructura es usada debido a que, como

dijimos anteriormente, se contempla el caso de que haya mas de un cliente en una calle, entonces la interfase devuelve estos clientes en orden de izquierda a derecha (esto es necesario por la forma que tiene el motor de cálculo de insertar a los clientes, partiendo una arista en dos). Cabe destacar que esta lista es auxiliar, y se borra de memoria una vez creado el archivo txt correspondiente, por lo que no influye en la memoria ocupada.

### Archivos .INF

Los shapefile, como mencionamos anteriormente, son un formato para representar información geográfica propiedad del ESRI. El mismo se compone de tres archivos: el .shp, que es el archivo donde se representa la información geográfica, el .dbf, que es una base de datos dbase con información ingresada por el usuario sobre la información geográfica del .shp, y un archivo .shx, que es un archivo de índice que actúa como nexo entre el .shp y el .dbf.

En nuestro software se hace un uso intensivo de la información del usuario sobre los datos geográficos (esto es, los datos que residen en el .dbf). Algunos ejemplos de estos datos son la demanda de un cliente, el costo de atravesar una calle, etc. El problema que se nos presentó es que un campo del archivo de base de datos dbase puede tener distintos nombres si pertenece a shapefiles distintos aun si representan lo mismo. Para solucionar este problema, y para brindar mas flexibilidad al usuario y mas independencia del software con respecto a los shapefiles, se exige que para cada shapefile usado exista un archivo de extensión INF donde se establezca los nombres de los campos del archivo dbase para dicho shapefile.

El formato de estos archivos .inf es el siguiente:

```
[Cabezal 1]
Nombre de campo 1
[Cabezal 2]
Nombre de campo 2
:
:
[Cabezal n]
Nombre de campo n
```

donde Cabezal representa un string fijo que representa una descripción del uso del campo.

A continuación detallamos los cabezales de los archivos .inf requeridos por el software:

- Para el shapefile de calles
  - Tiempo de cruce
- Para el shapefile de clientes
  - Demanda
  - Hora mínima
  - Hora máxima
  - Tiempo de atención
- Para el shapefile de depósitos
  - Hora mínima
  - Hora máxima
- Para el shapefile de clientes del DARP
  - Id del cliente
  - Hora
  - Tiempo de atención
  - Operación

### ***Estructuras de datos utilizadas para los algoritmos de ruteo***

ESQUINAS(Mapa):

Se representa mediante un array en el que cada elemento corresponde a una esquina y que tiene entre sus datos la indicación de si es un cliente o no, de si es un depósito o no, y la lista de esquinas adyacentes a la misma (lista de punteros). Se utiliza esta estructura pues las operaciones de insertar y consultar elementos son eficientes y además permite utilizar la memoria en forma adecuada. De esta forma logramos optimizar los tiempos y la utilización de memoria lo cual puede llegar a ser un problema dada la gran cantidad de datos que se deben mantener durante la ejecución de los algoritmos, por ej: grafo con las esquinas, rutas formadas, clientes, costos entre aristas, ahorros, etc.

Otra opción es realizar una matriz tal que el elemento ij indique si hay o no una “calle” o arista de la esquina i a la esquina j. Esta posibilidad es descartada pues para representar una ciudad de 15.000 esquinas serían necesarios alrededor de 225 Mb.

Además generalmente cada esquina es adyacente a un pequeño número de esquinas (cuatro aproximadamente) por lo que no es adecuada la representación mediante matrices.

Se realiza la clase GRAFO para el manejo de esta estructura brindando todas las funciones necesarias para ello.

#### RUTAS:

Las rutas que se van formando son almacenadas en un array en el que cada elemento corresponde a una ruta. Nuevamente utilizamos este tipo de estructuras para lograr un rápido acceso a los datos. Cada elemento del array contiene además del identificador de la ruta, el depósito al cual pertenece, la demanda de la ruta y la lista de clientes de la misma (lista de punteros). Existe un puntero al primer cliente y un puntero al último cliente de la misma ya que son los datos utilizados con más frecuencia y permiten una rápida unión de rutas al “conectar” el último cliente de una ruta con el primer cliente de otra ruta.

Se realiza la clase RUTA para el manejo de esta estructura brindando todas las funciones necesarias para ello.

#### CLIENTES Y DEPOSITOS:

Al recibir los datos de los clientes y depósitos entre la información brindada se encuentra las esquinas entre las cuales están ubicados. Para representar a los clientes y depósitos en la estructura que contiene el mapa se insertan como si fueran una nueva esquina que tiene como adyacentes las esquinas antes mencionadas. Esto se hace internamente para facilitar el ruteo pero el mapa original no es modificado.

Cada cliente y el depósito con sus datos correspondientes (demandas, time windows, etc.) son representados mediante un array y se identifican tomando en cuenta el índice del array.

#### AHORROS:

Para almacenar los ahorros correspondientes a cada par de clientes se utiliza un Heap Parcial, este heap consiste en lo siguiente: se almacenan en el heap los primeros  $h$  ahorros calculados hasta que el heap esté completo. Luego se compara cada uno de los siguientes ahorros calculados con el menor elemento del heap, si el nuevo elemento es mayor que el menor elemento del heap, se inserta este elemento quitando del heap el menor elemento. De esta manera se puede mantener en memoria los elementos que se utilizarán primero y si se necesitan nuevos elementos se actualiza el heap.

## **VRP**

### **Algoritmos**

Para resolver el VRP usaremos la modalidad más simple del Algoritmo de Clark y Wright.

### **Datos**

Como datos de entrada tenemos:

Archivo shapefile con la información de las calles: ubicación geográfica de cada arista, costo de recorrer la misma.

Archivo/s shapefile con la información de los clientes: ubicación geográfica, demanda.

Archivo shapefile con la información del depósito: ubicación geográfica.

Capacidad de los vehículos.

### **Diseño de alto nivel del algoritmo**

Recorrer Shapefile de arcos y cargar estructura que representa el grafo.

Recorrer Shapefile de clientes y depósito y cargar las estructuras con los datos correspondientes a clientes y depósito y agregar dichos puntos al grafo.

Generar estructura de datos que representa las rutas, de forma que haya un camino de ida y uno de vuelta a cada cliente desde el depósito.

Calcular ahorros según el algoritmo de Clark y Wright y guardarlos en el Heap.

Recorrer Heap y unir caminos según el algoritmo de Clark y Wright.

El Algoritmo termina cuando no hay más ahorros para considerar.

## **TSP**

El TSP es un caso particular del VRP. Si los datos ingresados corresponden a un TSP automáticamente se resolverá dicho problema, de lo contrario se resuelve el VRP. Para ejecutar el TSP los clientes deben tener demanda cero. La capacidad de los vehículos no es considerada pues al no tener demanda los clientes

## **MDVRP**

Podemos diferenciar dos instancias a la hora de resolver el MDVRP: Asignar que implica determinar a que depósito se va a relacionar cada cliente y Rutear que implica determinar las rutas de cada depósito a sus clientes.

## **Algoritmos**

Existen dos métodos: a) Asignar y rutear simultáneamente b) asignar y luego rutear.

Asignar y rutear simultáneamente no es aconsejable para problemas grandes, de más de 1000 clientes; por lo que no utilizamos dicha técnica pues estaríamos limitando fuertemente nuestro algoritmo.

Realizamos una variante del asignar y luego rutear clásico. Los clientes que se encuentren “notoriamente” mas cerca de un depósito son asignados al mismo, los que no, quedan marcados como clientes frontera. Al final se insertan los clientes frontera en las rutas ya formadas. Para ello utilizamos un valor al cual llamaremos radio, con  $0 < \text{radio} < 1$ .

## **Datos**

Como datos de entrada tenemos:

Archivo shapefile con la información de las calles: ubicación geográfica de cada arista, costo de recorrer la misma.

Archivo/s shapfile con la información de los clientes: ubicación geográfica, demanda.

Archivo shapfile con la información de los depósitos: ubicación geográfica, capacidad.

Radio

## **Diseño de alto nivel del algoritmo**

Recorrer Shapefile de arcos y cargar estructura que representa el grafo.

Recorrer Shapefile de clientes y depósitos y cargar las estructuras con los datos correspondientes a clientes y depósitos y agregar dichos puntos al grafo.

Para cada cliente:

Hallar el depósito más cercano D1 (distancia al depósito =  $d_1$ ) y el segundo depósito más cercano D2 (distancia al depósito =  $d_2$ )

Calcular el radio  $r = d_1/d_2$ . Si  $r < d$  asignamos el cliente al depósito D1, sino el cliente pasa a ser cliente frontera (mantener la información de D1 y D2).

Aplicar el algoritmo del VRP para cada depósito considerando los clientes asignados a cada uno.

Para cada uno de los clientes frontera insertarlo en las rutas del depósito D1 o D2 determinando el lugar de inserción como el que genera menor costo considerando todas las aristas de todas las rutas asignadas a los depósitos D1 y D2.

## **Consideraciones**

En la última etapa se insertan los nodos frontera en alguna de las rutas formadas para los dos depósitos más cercanos al mismo. Para ello se verifica cuál es el costo de insertar al cliente en cada arista de cada una de las rutas de los dos depósitos, insertando el cliente en el lugar que ocasiona menor incremento de costo, controlando que la demanda de la ruta no sobrepase la capacidad de los vehículos. Este método exhaustivo asegura un buen resultado para la inserción del cliente pero es muy costoso en cuanto a tiempo de ejecución por la gran cantidad de operaciones que deben realizarse.

Para mejorar la performance de esta etapa realizamos una estructura auxiliar en la que se cargan los clientes frontera ordenados por su depósito más cercano en primera instancia y por su segundo depósito más cercano en segunda instancia. Si para un cliente  $C_i$  tenemos que los depósitos más cercanos son D1 y D2 respectivamente, denotamos tal situación como  $C_i(D_1, D_2)$ . Si tenemos los siguientes nodos frontera:  $C_1(D_1, D_2)$ ,  $C_2(D_3, D_2)$ ,  $C_3(D_1, D_3)$ ,  $C_4(D_2, D_3)$ ,  $C_5(D_3, D_1)$  entonces en la estructura auxiliar utilizada los nodos frontera estarán en el siguiente orden:  $C_1, C_3, C_4, C_5, C_2$ . Esto nos permite por ejemplo trabajar simultáneamente con los clientes  $C_1$  y  $C_3$  sobre las rutas del depósito D1. A medida que se avanza en la recorrida de la estructura auxiliar se van insertando los clientes en los lugares seleccionados lo cual nos permite tener actualizada la demanda de las rutas formadas.

El valor del Radio es recibido como parámetro y variando el mismo pueden obtenerse distintos resultados y performance del algoritmo. Si se toma un valor muy bajo, 0.2 por ejemplo, gran cantidad de clientes quedará marcados como nodos frontera, lo cual como ya vimos aumenta considerablemente el tiempo de ejecución del algoritmo pero nos da la posibilidad (no nos asegura) de obtener seguramente un mejor resultado. Por otro lado tomando un valor muy alto, 0.9 por ejemplo, estaríamos prácticamente frente al método de primero asignar y después rutear el cual es claramente más rápido que el método anterior pero probablemente no de tan buenos resultados. Consideramos oportuno adaptar el valor del Radio a las necesidades de cada problema en particular siendo 0.7 un posible valor a tomar como estándar.

## ***VRPTW***

Resolvemos problemas con una ventana de tiempo pudiendo ser ampliado el algoritmo para aceptar mas de una ventana de tiempo.

### **Algoritmos**

Para resolver el VRPTW usaremos el Algoritmo de Clark y Wright adaptado para problemas con ventanas de tiempo.

### **Datos**

Como datos de entrada tenemos:

Archivo shapefile con la información de las calles: ubicación geográfica de cada arista, costo y tiempo que se demora en recorrer la misma.

Archivo/s shapefile con la información de los clientes: ubicación geográfica, demanda, ventana de tiempo.

Archivo shapefile con la información de los depósitos: ubicación geográfica, capacidad, ventana de tiempo.

Capacidad de los vehículos.

Costo de una unidad de tiempo. (Ver Consideraciones)

### **Diseño de alto nivel del algoritmo**

Recorrer Shapefile de arcos y cargar estructura que representa el grafo.

Recorrer Shapefile de clientes y depósitos y cargar las estructuras con los datos correspondientes a clientes y depósitos y agregar dichos puntos al grafo.

Generar estructura de datos que representa las rutas, de forma que haya un camino de ida y uno de vuelta a cada cliente desde el depósito.

Calcular ahorros según el algoritmo de Clark y Wright adaptado para problemas con ventanas de tiempo y guardarlos en el Heap.

Recorrer Heap y unir caminos según el algoritmo de Clark y Wright.

El Algoritmo termina cuando no hay más ahorros para considerar.

### **Consideraciones**

El costo de una unidad de tiempo es recibido como parámetro y se utiliza para uniformizar la medida del tiempo con la medida del costo asociado a una determinada arista del grafo.

## ***MDVRPTW***

En este caso aplicamos el método de asignar y luego rutear en forma pura sin la variante referida a los clientes frontera vista en el caso del MDVRP. En dicho caso los clientes que no están “claramente” más cerca de un cliente que del resto son marcados como clientes frontera y luego de finalizado el ruteo de los clientes que no están en dicha situación se insertan los clientes frontera en alguna de las rutas formadas para los 2 depósitos más cercanos al mismo.

Si dejamos clientes para insertar al final en rutas ya formadas, las ventanas de tiempo agregan una gran complejidad al problema y provocan un significativo aumento del tiempo de ejecución del algoritmo, un considerable aumento en la complejidad del algoritmo y puede deparar en realizar rutas exclusivas para los clientes fronteras. Dado que como ya vimos priorizamos el hecho de realizar algoritmos rápidos y reusables decidimos no utilizar la variante de los clientes frontera.

## Datos

Como datos de entrada tenemos:

Archivo shapefile con la información de las calles: ubicación geográfica de cada arista, costo de recorrer la misma.

Archivo/s shapfile con la información de los clientes: ubicación geográfica, demanda, ventana de tiempo.

Archivo shapfile con la información de los depósitos: ubicación geográfica, capacidad, ventana de tiempo.

## Pseudocódigo

Recorrer Shapefile de arcos y cargar estructura que representa el grafo.

Recorrer Shapefile de clientes y depósitos y cargar las estructuras con los datos correspondientes a clientes y depósitos y agregar dichos puntos al grafo.

Para cada cliente:

Hallar el depósito más cercano que sea factible y asignar el cliente a dicho depósito

Aplicar el algoritmo del VRPTW para cada depósito considerando los clientes asignados a cada uno.

## Consideraciones

El chequeo de factibilidad de asignar un cliente a un depósito consiste en lo siguiente: sea  $T_{maxCli}$  el tiempo máximo en que se puede llegar al cliente dada su ventana de tiempo,  $T_{maxDep}$  el tiempo máximo en que se puede llegar al depósito dada su ventana de tiempo,  $T$  el menor tiempo en que se puede ir desde el cliente al depósito; entonces si  $T_{maxCli} + T > T_{maxDep}$  la inserción no es factible, de lo contrario si lo es. Estamos haciendo un chequeo del peor caso pues consideramos el máximo tiempo en que se puede llegar al cliente. Al armar la ruta el tiempo final de llegada al cliente posiblemente sea menor al máximo permitido por la ventana de tiempo pero para realizar un chequeo confiable debemos tomar el peor caso. Aplicando este chequeo para cada cliente nos aseguramos que es factible llegar desde el último cliente al depósito a tiempo. La factibilidad de llegar de un cliente a otro en las rutas formadas se chequea en el VRPTW al formar las rutas.

## Flota Heterogénea

### Algoritmos

Para resolver el VRP con Flota Heterogénea usaremos el algoritmo Realistic Opportunity Savings (ROS) descrito en el capítulo anterior.

## Datos

Como datos de entrada tenemos:

Archivo shapefile con la información de las calles: ubicación geográfica de cada arista, costo de recorrer la misma.

Archivo/s shapefile con la información de los clientes: ubicación geográfica, demanda.

Archivo shapefile con la información del depósito: ubicación geográfica.

Archivo de texto con la información de los vehículos: cantidad y capacidad. (Ver Consideraciones)

## Pseudocódigo

Recorrer Shapefile de arcos y cargar estructura que representa el grafo.

Recorrer Shapefile de clientes y depósito y cargar las estructuras con los datos correspondientes a clientes y depósito y agregar dichos puntos al grafo.

Generar estructura de datos que representa las rutas, de forma que haya un camino de ida y uno de vuelta a cada cliente desde el depósito.

Calcular ahorros según el algoritmo Realistic Opportunity Savings (ROS) y guardarlos en el Heap.

Recorrer Heap y unir caminos según el algoritmo Realistic Opportunity Savings (ROS).

El Algoritmo termina cuando no hay más ahorros para considerar.

## Consideraciones

El archivo de texto que contiene la información de los vehículos consiste en una línea por cada tipo de vehículo; el tipo de vehículo está asociado a la capacidad de cada vehículo. Cada línea del archivo contiene la capacidad de ese tipo de vehículo y la cantidad de vehículos de ese tipo.

Al comienzo del algoritmo se supone que existe un vehículo para cada cliente y se asigna a cada cliente un vehículo tal que su capacidad sea la mínima indispensable para atender a ese cliente.

## **CPP**

En primera instancia realizamos el siguiente análisis sobre el CPP.

El problema del cartero chino consiste en, dado un grafo con costos asociados a sus aristas, hallar un ciclo que pase al menos una vez por cada arista, minimizando el costo total.

Sea  $G$  el grafo de entrada (suponemos que previamente se seleccionaron algunas aristas del grafo “ciudad”). La idea del algoritmo que presentamos es la siguiente: si un grafo contiene un ciclo euleriano, entonces dicho ciclo es la solución buscada (a la que llamaremos CPT, o chinese postman tour). Si no lo es, transformaremos el grafo  $G$  en un nuevo grafo  $G'$  euleriano repitiendo aristas de  $G$ , de tal forma que la cantidad de aristas agregadas sea de costo mínimo.

Los pasos para hallar el CPT son los siguientes:

- 1) Chequear que el grafo  $G$  sea fuertemente conexo (de otra forma, el algoritmo funcionará incorrectamente)
- 2) Usar el algoritmo de Dijkstra para obtener los costos y los caminos mas cortos entre todo par de vértices de  $G$ .
- 3) Identificar  $N'$ , que es el conjunto de los vértices tal que su grado es impar.
- 4) Del conjunto de todos los caminos posibles obtenidos en el paso anterior, hallar el subconjunto de menor costo tal que todos los vértices de  $N'$  sean ‘usados’ (dicho de otra forma, obtener una bipartición del grafo generado por  $N'$ ).
- 5) Agregar los caminos obtenidos en 3) al grafo  $G$ , obteniendo el grafo  $G'$ . Notar que  $G'$  es euleriano.
- 6) Determinar el circuito euleriano de  $G'$ . Este es el CPT buscado.

Ahora explicaremos detalladamente la implementación de cada uno de los pasos:

- 1) Este paso se hace junto con el paso 2. Si algún camino mas corto en Dijkstra tiene valor infinito, significara que el grafo no era conexo.
- 3) Como el grafo va a estar representado como una lista de aristas, lo que se debe hacer es recorrer, para cada nodo, sus aristas.
- 4) Este paso lo vamos a hacer usando backtracking
- 6) El procedimiento es el siguiente: construimos un árbol abarcador de  $G$ , y luego escogemos las aristas de  $G'$  de tal forma de no tomar aristas pertenecientes al árbol si alguna otra arista (que parta del nodo en donde estamos parados) esta disponible.

Posteriormente en una reunión con el usuario (2/9/99) se establece que el problema a resolver no será el CPP clásico, luego de plantearse como será recibida la información con la demanda en los arcos.

Se determina que en cada arco pueden haber varios clientes con demanda y que la información brindada es la demanda de cada cliente y no la demanda total del arco. Con esto el problema queda planteado como satisfacer la demanda de todos los clientes los cuales están ubicados en los arcos sin necesidad de recorrer completamente todos los arcos.

Para ello transformamos el mapa inicial agregando una esquina en la posición de cada cliente y luego aplicamos los algoritmos vistos para los casos en que la demanda se encuentra en los clientes.

## **DARP**

Resolvemos el problema de DARP con un solo depósito, vehículo con capacidad limitada, pickup y delivery de 1 persona por lo que la demanda de cada cliente es 1, tiempo mínimo para pickup y máximo para delivery.

Este es un caso básico del DARP y las variantes mas complicadas escapan a los objetivos de este proyecto.

Las restricciones de tiempo se consideran de la siguiente manera: si el cliente  $1$  tiene una ventana de tiempo  $[t_1, t_2]$  ello significa que se debe llegar a recoger al cliente en el tiempo  $t_1$  o más (no se puede llegar antes y quedar esperando) y debe ser llevado a destino antes de tiempo  $t_2$ . La restricción de no poder llegar antes de tiempo a recoger el cliente y esperar es por una cuestión de imagen de la empresa pues se considera que puede haber otros pasajeros en el vehículo en ese momento y no deben esperar mientras se recoge a otro pasajero. Siempre debe pasarse a buscar a los clientes lo antes posible.

Supongamos por un instante que tenemos cinco clientes numerados del 1 al 5. Tendremos cinco puntos de pickup ( $P_1 P_2 P_3 P_4 P_5$ ) y cinco puntos de delivery ( $D_1 D_2 D_3 D_4 D_5$ ). Sean  $(I_1 F_1)$ ,  $(I_2 F_2)$ , etc. las restricciones de tiempo para los clientes 1, 2, etc. Sea  $T_i$  el tiempo mínimo en que se puede ir de los puntos  $P_i$  a los puntos  $D_i$

Resolvemos el problema de la siguiente forma:

Primero formar un VRPTW con los puntos P1 P2 P3 P4 P5 considerando las restricciones de capacidad del vehículo y restricciones de tiempo. Dado un cliente  $i$  las restricciones de tiempo a considerar en el VRPTW para el punto  $P_i$  son:  $(I_i F_i - T_i)$ . Resulta evidente que el tiempo mínimo en que se puede pasar por el punto  $P_i$  es  $I_i$  dado que es la restricción original del cliente  $i$ . Sean  $T(P_i)$  y  $T(D_i)$  el tiempo en que el vehículo pasa por el punto  $P_i$  y  $D_i$  respectivamente.

Afirmamos que  $T(P_i)$  debe ser menor o igual que  $F_i - T_i$  pues:

Si  $T(P_i) > F_i - T_i$  entonces  $T(P_i) + T_i > F_i$ .

Como  $T_i$  es el tiempo mínimo para ir de  $P_i$  a  $D_i$  entonces  $T(D_i) \geq T(P_i) + T_i$

Entonces  $T(D_i) > F_i$  lo cual significa que el vehículo pasa por el punto  $D_i$  después del tiempo  $F_i$  que era el máximo permitido por las restricciones de tiempo del cliente.

Luego se insertan en la ruta ya formada cada uno de los puntos  $D_1 D_2 D_3 D_4 D_5$ , en forma ordenada, sin volver atrás en la solución y minimizando el tiempo utilizado. Al decir en forma ordenada nos referimos a que primero insertamos el  $D_1$ , luego el  $D_2$ , etc. Al decir sin volver atrás nos referimos a que una vez formado el TSPTW con los puntos  $P_1 P_2 \dots P_5$  dicho orden no se modifica y lo mismo ocurre después de cada inserción de los puntos de entrega en la ruta, es decir, para cada caso se busca la mejor solución posible dada la ruta ya formada sin considerar volver a formar la ruta de otra forma.

En acuerdo con el usuario se asume que los datos recibidos permiten desarrollar el algoritmo antes descrito. Es decir no puede ocurrir por ejemplo que:

a)  $I_i + T_i \leq F_i$

b)  $I_1=8:00 F_1=8:30 I_2=8:20 F_2=8:50$ . Supongamos que el mínimo tiempo que toma ir de un punto cualquiera a otro es 0:20, entonces el problema no tiene solución con el algoritmo propuesto.

Esto es porque en la primera etapa (TSPTW) se determina que se debe pasar primero por el punto  $P_1$  y luego por el punto  $P_2$ . Como el tiempo máximo de llegada al punto  $D_1 (F_1)$  es 8:30 el punto  $D_1$  debe insertarse entre  $P_1$  y  $P_2$  pues si se lo inserta luego de  $P_2$  se llegaría a  $D_1$  no antes de 8:40. Finalmente tenemos que el punto  $D_2$  debe insertarse luego de  $P_2$  por definición del problema, pero al punto  $P_2$  se llega no antes de 8:40 por lo que al punto  $D_2$  se llega no antes de las 9:00 lo cual viola la restricción de tiempo para dicho punto.

## 5 Testeo

El testeo o testing se utiliza para verificar que un programa se comporte como fue especificado, tiene como fin principal aumentar la confiabilidad en las cualidades de un sistema y debe estar basado en técnicas sistemáticas. El testeo debe ayudar no solo a detectar errores sino también a localizarlos y debe ser repetible, es decir que debe ser posible que al repetir un experimento con la misma entrada y el mismo código se produzcan iguales resultados.

El desarrollo del sistema fue dividido en módulos claramente diferenciados. Una primera división podemos establecerla en: a) desarrollo de la interfase gráfica y b) desarrollo de los algoritmos que resuelven los problemas de ruteo. A su vez el desarrollo de los algoritmos también fue dividido en varios módulos entre los que se encuentran: a) funciones para entrada de datos y carga de las estructuras de datos a utilizar, b) funciones para generación del resultado a partir de los datos obtenidos en los algoritmos, c) resolución de VRP, d) resolución de MDVRP, etc. Debemos tener en cuenta que muchos de estos módulos “utilizan” funcionalidad brindada por otros módulos, por ejemplo para resolver el MDVRP se necesita resolver el VRP. Finalmente se realizó un módulo, al que llamaremos módulo principal, que es el que recibe los datos desde la interfase gráfica y realiza todas las operaciones necesarias utilizando las funcionalidad brindada por los otros módulos.

Un importante punto a considerar a la hora de establecer el testeo a realizar es la forma en que son ingresados los datos al sistema. Según lo acordado con el usuario los datos son entregados en archivos .shp. La generación de dichos archivos con la información geográfica de calles, clientes y depósitos no es simple y requiere de herramientas especiales (por ej. ArcView, software del ESRI) y conocimiento de las mismas por lo que acordamos con el usuario que nos fueran entregados datos para realizar las pruebas. Con estos datos buscamos abarcar los diferentes casos posibles realizando pruebas de pequeño y mediano tamaño (mapas de 2000 esquinas, de 2 a 50 clientes) para poder analizar el comportamiento general del sistema. El problema que se nos presenta con estos datos es que no tenemos una solución documentada de cual es el resultado óptimo. Para los casos chicos nosotros calculamos la solución óptima pero para los casos grandes (2000 esquinas y 50 clientes) dicho cálculo no está a nuestro alcance.

Por otro lado obtuvimos datos de bibliotecas en internet (17) los cuales no contaban con la información geográfica de las calles (mapa) pero si las coordenadas de ubicación de los clientes y depósitos o las distancias entre ellos, con lo que simulamos un mapa que cuenta sólo con dichos puntos (clientes y depósitos) como esquinas. En este caso si tenemos el valor de la solución óptima por lo que tenemos un parámetro real para comparar con nuestros resultados pero al no tener datos geográficos no podemos testear la interacción con la interfase gráfica.

El conjunto de datos brindados por el usuario es el utilizado durante el desarrollo de los diferentes módulos. Podemos diferenciar dos tipos de testeo: white-box testing o testeo de caja blanca (se basa en la estructura interna de los módulos) y black-box testing o testeo de caja negra (sin tener conocimiento de la estructura interior del programa, solo se basa en las especificaciones). Utilizamos las dos estrategias ya que son complementarias, mientras white-box se ocupa de lo que hace, black-box se ocupa de lo que tendría que hacer el programa.

En cuanto al testeo de caja blanca utilizamos el criterio de cubrimiento de sentencias el cual se basa en que cada sentencia elemental sea ejecutada al menos una vez. Para ello utilizamos casos de prueba pequeños y variados que permiten su fácil seguimiento (mediante debugging por ejemplo) y cubrir las diferentes situaciones posibles.

Para el testeo de caja negra formamos diferentes conjuntos de datos y estimamos cual debería ser la salida de los mismos. Se ejecuta el programa y se observa la salida del mismo sin entrar en detalle de cómo se llegó a dicho resultado. Se compara la salida obtenida con la prevista determinando si la prueba se toma como válida o se encuentra algún error. Realizamos reiteradas pruebas con similares datos de entrada con pequeñas variaciones (capacidad de los vehículos por ej.) pudiendo analizar de esta forma como se comporta el sistema frente a cambios puntuales.

En cuanto al testeo de integración de los módulos el mismo se fue haciendo en forma simultánea al testeo de los módulos por separado. Esto se debe a que por ejemplo el módulo que resuelve el VRPTW utiliza funcionalidad del módulo del VRP por lo que al testear el módulo del VRPTW se está testeando también su integración con el módulo del VRP. Finalmente al testear el módulo principal del sistema estamos testeando la integración de la interfase gráfica con cada uno de los módulos que resuelven los distintos problemas de ruteo.

Los datos obtenidos de bibliotecas de internet (17) son utilizados para el testeo final, realizado una vez terminados los módulos. Dado que el formato de dichos datos es diferente al especificado por el usuario adaptamos los módulos realizados para poder realizar el testeo con estos datos.

Conseguimos datos que tuvieran información sobre la mejor solución posible (en cuanto a distancia total recorrida) sólo para el caso del VRP. Dichos datos son utilizados para testear la “eficacia” de la heurística utilizada, además del funcionamiento

general del sistema (generación correcta de rutas, cubrimiento de todos los clientes, etc) lo cual es testado en todas las pruebas realizadas.

Para el caso del DARP no conseguimos datos en internet para realizar las pruebas pues lo que hicimos fue un caso particular del DARP. Por lo tanto el DARP fue testado solamente con los datos entregados por el usuario.

Las pruebas fueron realizadas en un PC con la siguiente configuración:

- Procesador Pentium MMX 75Mhz
- 40 Mb RAM

En la siguiente tabla vemos las características y resultados de las pruebas realizadas.

Tipo de problema	Dimensión (n° de clientes)	Mejor Valor según los datos	Nuestro Resultado	Tiempo de ejecución (segundos)
VRP	22	585	716.2157	3
VRP	23	875	945.3251	3
VRP	30	545	552.7412	4
VRP	33	910	943.0978	4
VRP	51	521	584.6373	4
VRP	76	847	914.2711	4
VRP	76	1058	1064.5992	4
VRP	76	745	794.7405	4
VRP	76	692	751.6903	4
VRP	101	1114	1143.4487	5
VRP	13	290	290	3
VRP	31	1212	1263	4
VRP	7	114	119	3
MDVP	12		250	4
MDVRP	30		1108	5
MDVRP	6		95	4
VRPTW	13		320	4
VRPTW	31		1412	5
VRPTW	7		142	4

Como podemos apreciar los resultados son en general apenas superiores a los presentados como mejor solución posible. No haber hallado la mejor solución se justifica pues no utilizamos algoritmos exactos sino algoritmos heurísticos los cuales nos aproximan a la mejor solución. Vemos que dicha aproximación es muy buena.

## 6 Dificultades

En esta sección veremos problemas que surgieron en el desarrollo del proyecto y retrasaron la marcha del mismo. Estos problemas no estaban previstos en la planificación del proyecto por lo que su impacto fue mayor al provocado por otras situaciones si previstas como por ejemplo correcciones a los algoritmos luego del testeo.

### *MapObjects*

Según lo acordado con el usuario los datos geográficos son entregados en archivos .shp (formato de ESRI), y para mostrar dichos datos y mostrar la solución hallada se utiliza la herramienta MapObjects brindada por el usuario. La interfase es realizada en Visual Basic utilizando MapObjects para la representación geográfica de los datos por lo que debimos investigar la utilización de MapObjects desde Visual Basic.

En los primeros días de mayo de 1999 fue entregado MapObjects por el usuario. Comenzamos el aprendizaje de cómo se utiliza el mismo y de cómo interactúa con Visual Basic. Surgieron diferentes problemas y lo planteado en los manuales e instructivos de MapObjects que conseguimos (18) no funcionaba. Luego de largas investigaciones concluimos que lo que nos habían entregado era para desarrollo en internet, era MapObjects Internet Map Server.

Nos fue entregada por el usuario una nueva “versión” del MapObjects pero nuevamente tuvimos problemas. En este caso el problema era que muchas de las funcionalidades mencionadas en los manuales no estaban disponibles. A fines de junio de 1999 (ver Control de Reuniones, apéndice 2) planteamos este problema al usuario quien a los pocos días nos entregó la versión definitiva de MapObjects.

Si bien simultáneamente realizamos otras tareas como la elección de los algoritmos a utilizar para resolver los problemas de ruteo y elección de estructuras de datos, durante estas semanas nuestra mayor atención estuvo centrada en el funcionamiento del MapObjects por lo que se produjo un importante retraso en el avance del proyecto.

El plan del proyecto (ver Plan de Trabajo y Avance del Proyecto, apéndice 3) marcaba el aprendizaje de MapObjects y realización de la interfase gráfica para el mes de junio lo cual se vio seriamente afectado por lo antes visto. El Plan de Trabajo inicialmente establecía realizar primero la interfase gráfica y luego la resolución de los algoritmos pero finalmente fue desarrollado todo en forma simultánea

### *Generación de dll*

Para llamar a las funciones que resuelven cada uno de los algoritmos vistos en el proyecto debimos generar un archivo .dll (Dynamic Link Library) que exportara dichas funciones. Este archivo debería ser generado en Borland C++ 5.0 para Windows 95, que fue la herramienta que elegimos para programar los algoritmos que resuelven los distintos problemas de ruteo tratados en el proyecto. En principio, nos aconsejaron no usar el compilador de Microsoft, y fue por eso que resolvimos utilizar el compilador de Borland. Cada una de las funciones fueron desarrolladas en Borland C++ 5.0 sin tener demasiados problemas, aunque tuvimos que dedicar algo de tiempo para aprender a usar esta herramienta pues las versiones de C++ que habíamos utilizado en ocasiones anteriores eran bajo DOS o UNIX. Cuando llegó el momento de generar el archivo .dll, comenzaron a surgir una serie de inconvenientes.

Primero hubo problemas porque estábamos intentando crear el archivo en un camino de directorios que contenía nombres largos, entonces el compilador devolvía un mensaje de error que no era muy claro pero igual generaba el archivo .dll. Al intentar usar este archivo, llamando a las funciones desde Visual Basic, vimos que no encontraba esas funciones como exportadas en la dll, entonces presumimos que el problema era el error que nos devolvía el compilador. Luego de buscar en la ayuda de Borland C++ 5.0 y en el sitio Web de Borland (17), no encontramos nada que hablara de problemas con los nombres largos, por lo tanto hasta ese momento ni se nos ocurrió que el problema era ese. Recordamos que cuando desarrollábamos la solución de los algoritmos nos había ocurrido algo parecido y al probar generar la dll en un camino de directorios que no contenía nombres largos, no hubo ningún error. Entonces probamos llamar a las funciones desde Visual Basic con la nueva dll generada pero volvió a devolver el mismo mensaje de error.

En ese momento decidimos consultar al Ing. José Comas sobre lo que nos estaba ocurriendo, quien nos dijo que el usaba el compilador de Microsoft y nos dijo que consultáramos a otro estudiante de Taller V que el conocía y que usaba el compilador de Borland, intentamos contactar a ese estudiante pero él ya había terminado el taller. Entonces mientras seguíamos buscando la solución en la página Web de Borland (17), tuvimos que buscar una solución alternativa que no era tan prolija pero que funcionaba y resolvía los problemas de ruteo que es el objetivo del taller. Esta solución consistía en generar un archivo .exe por cada una de las funciones que eran exportadas en la dll.

Finalmente en la primer reunión realizada en el mes de febrero, el Ing. José Comas nos comentó que el había tenido algún problema cuando comenzó a utilizar el compilador de C++, el problema era que las funciones exportadas por la dll tenían un

nombre distinto al original. Intentamos utilizar el comando que nos dio José para resolver este problema pero en Borland este comando tiene otra función. A partir de ese momento sabíamos cual era el problema puntual, entonces resultaba más fácil buscar en la página Web de Borland y después de algunos días encontramos la solución (un mes y medio después de generada la primer dll). El problema estaba en la sintaxis utilizada al declarar las funciones a ser exportadas.

## 7 Resultados y conclusiones

Se desarrolla un paquete de software genérico de acuerdo a los requerimientos del usuario. Se resuelven los problemas básicos y las extensiones mas significativas.

Vemos nuevamente la tabla presentada al comienzo donde se puede apreciar los casos que resuelven algunos paquetes comerciales muy difundidos y agregamos los casos resueltos por nuestro sistema.

	TSP	TSP TW	TSP FH	VRP	VRP TW	VRP FH	MDVRP	CPP	DARP
DynaRoute	-	-	-	-	-	-	-	-	Si
GeoRoute 5	Si	-	-						
GeoPostal	-	-	-	-	-	-	-	Si	-
RouteSmart	Si	-	-						
SHIPCONS II	Si	-	-						
TransCad	Si	-							
Trapeze-Taxi	-	-	-	-	-	-	-	-	Si
Direct Route	Si	Si	Si	Si	Si	Si	S/i	-	-
GeoRoute-Municipal	-	-	-	-	-	-	-	Si	-
LoadExpress Plus	Si	-	-						
RiMMS	Si	s/i	s/i	Si	s/i	s/i	S/i	-	-
RoutePro Dispatcher	Si	-	-						
Routronics 2000	-	-	-	-	-	-	-	-	Si
STARS	Si	-	-						
Optrak	Si	s/i	s/i	Si	s/i	s/i	Si	-	-
TruckStops for Windows	Si	-	-						
ArcLogistics Route	Si	-	-						
<b>Nuestro sistema</b>	<b>Si</b>								

Destacamos que nuestro sistema es el único que resuelve todos los casos aquí mencionados. De esta forma se logra una gran variedad y generalidad que era el principal objetivo de este proyecto. Vemos por ejemplo que todos los sistemas que resuelven el CPP o el DARP son “especializados” y resuelven solo ese tipo de problemas.

Lograr un paquete que resuelva todos los casos de ruteo de vehículos es casi imposible ya que los distintos problemas que se pueden formular modificando las restricciones y/o objetivos buscados son prácticamente infinitos. Por ello se desarrollaron los casos básicos y las principales extensiones de ellos cubriendo de esta forma un amplio espectro de problemas.

Los resolución de los diferentes problemas fue realizada modularizando lo más posible los algoritmos y estructuras de datos utilizadas permitiendo de esta forma una fácil adaptación a futuras variantes de los problemas resueltos. Son totalmente independientes el ingreso de datos (datos geográficos, clientes, depósitos, etc), almacenamiento de datos, resolución de algoritmos, salida de datos, etc. Con ello se logra que por ejemplo para modificar el ingreso de datos sólo sea necesario modificar la interfase realizada sin necesidad de saber como funcionan el resto de los módulos. De esta manera este sistema puede ser utilizado fácilmente como base para futuros desarrollos.

Como trabajo futuro a realizar sobre este proyecto puede considerarse:

- Mejoras en la interfase gráfica permitiendo mayor flexibilidad en la presentación de los resultados. El objetivo del proyecto está basado en la resolución de los problemas de ruteo y no en producir un paquete de software para ser comercializado lo cual explica que la interfase gráfica no haya sido desarrollada en profundidad. Se puede mostrar por ej. una lista con los clientes asignados a cada ruta, hora de partida para los casos con restricciones de tiempo, más opciones en la barra de herramientas para el manejo gráfico de los datos, etc.
- Adaptar los algoritmos para resolver otro tipo de problemas o variantes de los problemas resueltos. Se pueden buscar mejores heurísticas que proporcionen mejores resultados para determinado tipo de problemas (menos de 100 clientes) las cuales no fueron utilizadas para no perder generalidad y flexibilidad en los algoritmos desarrollados.
- Conexión a un SIG (Sistema de Información Geográfica) más potente (Arcview por ej.) que permita trabajar directamente con los datos geográficos y no sólo utilizarlos para entrada y salida de datos, lo cual escapa al alcance de este proyecto.

## 8 Referencias

- [1] Software Library for Operations Research - University of Karlsruhe  
[http://www.wior.uni-karlsruhe.de/Bibliothek/Software\\_for\\_OR/Vehicle\\_Routing/com/aacom.html](http://www.wior.uni-karlsruhe.de/Bibliothek/Software_for_OR/Vehicle_Routing/com/aacom.html) (4/99)
- [2] SAITECH, Inc.  
<http://www.saitech-inc.com> (4/99)
- [3] GIRO Enterprises, Inc.  
<http://www.giro.ca> (4/99)
- [4] RouteSmart Technologies  
<http://www.routesmart.com> (4/99)
- [5] Insight, Inc.  
<http://www.insight-mss.com> (4/99)
- [6] Caliper Corporation  
<http://www.caliper.com> (4/99)
- [7] Trapeze Software Group  
[www.trapezesoftware.com](http://www.trapezesoftware.com) (4/99)
- [8] Appian Logistics Software  
<http://www.appianlogistics.com> (4/99)
- [9] Prophecy Transportation Software, Inc.  
<http://www.mile.com/loadexpress.htm> (4/99)
- [10] Lightstone Group, Inc.  
<http://www.lightstone.com> (4/99)
- [11] CAPS Logistics, Inc.  
<http://www.caps.com> (4/99)
- [12] Carrier Logistics, Inc.  
<http://www.carrierlogistics.com> (4/99)
- [13] Optrak Distribution Software Ltd  
<http://www.optrak.co.uk> (4/99)
- [14] Micro Analytics, Inc  
<http://www.bestroutes.com> (4/99)
- [15] Environmental Systems Research Institute, Inc.
- [16] <http://www.zib.de> - Software - eLib - archive – MP Testdata - Vehicle Routing – VRPLIB – MP TESTDATA  
<ftp://ftp.zib.de/pub/Packages/mp-testdata/tsplib/vrp/index.html> (12/99)
- [17] <http://www.borland.com/bcppbuilder/news/publications.html> (12/99)
- [18] <http://www.esri.com/software/mapobjects/index.html> (6/99)
  
- [a] “Vehicle Routing: Methods and Studies” Bruce Golden, Arjang Assad
- [b] “Routing and scheduling of vehicles and crews” Lawrence Bodin, Bruce Golden, Arjang Assad y Michael Ball

## 9 Apéndices

- Apéndice 1: Soluciones Existentes Actualmente.
- Apéndice 2: Control de Reuniones.
- Apéndice 3: Plan de Trabajo y Avance del Proyecto.

## ***Soluciones Existentes Actualmente.***

En la siguiente Sección veremos un resumen de las principales funcionalidades de varios paquetes comerciales para resolver el problema de ruteo de vehículos. La siguiente lista fue obtenida de la biblioteca de software de investigación de operaciones de la Universidad de Karlsruhe [1].

### **DynaRoute**

DynaRoute es un sistema de ruteo de vehículos y scheduling para el manejo de despacho en tiempo real de vehículos. El sistema determina asignaciones óptimas entre vehículos y nuevas órdenes tomando en cuenta la posición del vehículo, horas de disponibilidad del vehículo y horas de disponibilidad del cliente. [2]

### **GeoRoute 5**

GeoRoute 5 usa una base de datos para representar la red de calles en forma detallada junto con un conjunto de algoritmos de ruteo especializado que producen eficientes rutas punto a punto. Puede ser usado diariamente o periódicamente para generar rutas basadas en pedidos de clientes o servicios periódicos. En ambos casos, GeoRoute 5 ayuda a reducir los costos operativos y a respetar los compromisos con los clientes.

Para cada cliente a ser visitado, se puede especificar distintos tipos de vehículos, cantidad que será entregada o retirada, tiempo que demorará el servicio y la posibilidad de ingresar ventanas de tiempo. [3]

### **GeoRoute Postal**

El producto GeoRoute Postal provee organización postal con funciones adicionales. Esto incluye la habilidad de planificar y optimizar la entrega de correspondencia como también operaciones motorizadas tales como recolección de mailbox, distribución de paquetes, etc.

Este sistema permite también definir los recursos disponibles y los modos de viaje (a pie, bicicleta, auto, etc.). Para cada modo, se puede especificar la velocidad de viaje y si las restricciones viales (calles flechadas, por ej.) deben ser respetadas o no. Los modos pueden combinarse, como en el caso que se use un vehículo para trasladarse a determinada zona y luego el reparto se haga a pie. [3]

### **RouteSmart**

RouteSmart evalúa las diversas formas en las cuales el conductor (o conductores) pueden viajar a través de las calles y servir a los clientes en su ruta, teniendo en cuenta diversas restricciones operacionales como capacidad de los vehículos, horario de servicio de los clientes, calles flechadas, etc. [4]

### **SHIPCONS II**

SHIPCONS II es un sistema designado para asistir a aquellas personas que deben planificar el despacho de productos para tomar mejores decisiones acerca de los recursos de transporte.

SHIPCONS II recibe información acerca de las órdenes que serán despachadas y los vehículos disponibles (con su capacidad, costo, etc.). Esta información es procesada para generar un cierto número de rutas factibles, las cuales son la entrada para un optimizador que seleccionará el conjunto de rutas que aseguren que todas las órdenes serán despachadas al menor costo. [5]

### **TransCAD**

TransCAD es un sistema de información geográfica diseñado específicamente para ser usado por profesionales del área de transporte para almacenar, mostrar, manejar y analizar datos de transporte.

TransCAD puede ser usado para todos los modos de transporte, a cualquier escala o nivel de detalle.

Este sistema provee un poderoso sistema de información geográfica con extensiones especiales para el área de transporte, herramientas de mapeo y visualización diseñadas para aplicaciones de transporte y módulos de aplicación para ruteo. [6]

### **Trapeze-Taxi**

Este sistema es una aplicación de ruteo y manejo diseñada para compañías de taxi.

Con Trapeze-taxi se puede despachar taxis en forma instantánea y notificar a los conductores los cambios, construir una base de datos de clientes con información estadística, generar reportes, etc. [7]

### **Direct Route**

El poderoso motor de ruteo de DirectRoute diseña rutas basadas en la ubicación del cliente, capacidad de los vehículos y horarios del cliente. Esto asegura el aprovechamiento máximo de los vehículos, la minimización de la distancia recorrida y la puntualidad del servicio. [8]

### **GeoRoute-Municipal**

Geo-Route Municipal es un poderoso software de ruteo que está específicamente diseñado para la creación automática de rutas para servicios en las ciudades, tales como recolección de residuos, limpieza de calles, remoción de nieve, etc. El software provee todas las herramientas necesarias para preparar los datos, crear y optimizar las rutas y mostrar los resultados en forma de mapas o reportes. [3]

### **LoadExpress Plus**

LoadExpress Plus for Windows ofrece una forma simple, poderosa y flexible de construir y optimizar rutas para la realización de entregas a clientes. Este sistema soporta el ingreso de restricciones tales como tamaño variable de los vehículos, horarios de entrega, etc. [9]

### **RiMMS**

El software de ruteo y scheduling RiMMS permite a las empresas manejar mejor sus recursos de transporte, mejorando la productividad y la eficiencia, reduciendo los costos operacionales.

RiMMS ayuda a los usuarios a diseñar rutas de entrega y asignar tareas a los recursos. El sistema integra tecnología de mapeo para crear rápidamente mapas a color con instrucciones de ruta precisas, basándose en varios parámetros de la base de datos del usuario. [10]

### **RoutePro Dispatcher**

RoutePro Dispatcher permite construir rutas a partir de la información de los clientes o adaptar rutas existentes a fluctuaciones diarias.

Entre otros parámetros, es posible definir horarios para cada cliente, especificar prioridades para los mismos, definir distintos tipos de vehículos con costos asociados, especificar horarios para los vehículos, etc. [11]

### **Routronics 2000**

El sistema Routronics 2000 es un despachador en tiempo real de vehículos. Las ventajas de su uso radican en el ahorro de tiempo y combustible, así como la satisfacción del cliente.

Routronics 2000 además brinda un seguimiento continuo de los conductores. Cuando el despachador recibe un pedido, se le brindan distintas rutas posibles. Este hace su elección, y la información va directamente al conductor. Los pedidos no asignados, o no confirmados, son mostrados en pantalla para que el despachador tome la acción correspondiente. [12]

### **STARS**

STARS (Smart Truck Assignment and Routing System) determina cargas de productos y rutas óptimas para vehículos de entrega. Como una herramienta de uso diario, STARS puede reducir substancialmente los costos operativos minimizando las distancias recorridas y el tiempo. [2]

### **Optrak**

Optrak reduce los costos de distribución planificando eficientemente las cargas y las rutas de los vehículos, haciendo el mejor uso de los recursos. Sus sofisticadas optimizaciones dan como resultado mejores soluciones a una amplia variedad de problemas. [13]

## Truckstops for Windows

Truckstops es un programa de ruteo que recibe tres tipos de dato:

- información de las paradas a realizar: clientes con la cantidad de bultos que se recogerán o entregarán, horarios, etc.
- información de los vehículos: costos, capacidad máxima
- información general: por ejemplo, cantidad de depósitos.

[14]

## ArcLogistics Route

ArcLogistics Route es un sistema que resuelve problemas complejos de ruteo de vehículos y scheduling. Este sistema determina que vehículo debe servir a cada cliente, y la mejor secuencia de servicio para cumplir con los requisitos temporales de los mismos, minimizando la distancia (o tiempo) recorrida. [15]

A continuación recordamos la tabla que compara los casos que resuelven los paquetes comerciales mencionados anteriormente.

	TSP	TSP TW	TSP FH	VRP	VRP TW	VRP FH	MVRP	CPP	DARP
DynaRoute	-	-	-	-	-	-	-	-	Si
GeoRoute 5	Si	Si	Si	Si	Si	Si	Si	-	-
GeoPostal	-	-	-	-	-	-	-	Si	-
RouteSmart	Si	Si	Si	Si	Si	Si	Si	-	-
SHIPCONS II	Si	Si	Si	Si	Si	Si	Si	-	-
TransCad	Si	Si	Si	Si	Si	Si	Si	Si	-
Trapeze-Taxi	-	-	-	-	-	-	-	-	Si
Direct Route	Si	Si	Si	Si	Si	Si	S/i	-	-
GeoRoute-Municipal	-	-	-	-	-	-	-	Si	-
LoadExpress Plus	Si	Si	Si	Si	Si	Si	Si	-	-
RiMMS	Si	s/i	s/i	Si	s/i	s/i	S/i	-	-
RoutePro Dispatcher	Si	Si	Si	Si	Si	Si	Si	-	-
Routronics 2000	-	-	-	-	-	-	-	-	Si
STARS	Si	Si	Si	Si	Si	Si	Si	-	-
Optrak	Si	s/i	s/i	Si	s/i	s/i	Si	-	-
TruckStops for Windows	Si	Si	Si	Si	Si	Si	Si	-	-
ArcLogistics Route	Si	Si	Si	Si	Si	Si	Si	-	-

De la biblioteca de software de investigación de operaciones de la Universidad de Karlsruhe [1] obtuvimos la especificación de varios productos (entre los cuales figuran los vistos anteriormente) de los cuales presentamos la página de internet relacionada a cada uno de ellos.

<b>Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a></b>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>CATRIN</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	<p>CATRIN consists of two elements: the tour-planning program and the digital road network. In the tour-planning process CATRIN considers the following factors:</p> <ul style="list-style-type: none"> <li>• road connections and driving time</li> <li>• vehicle-capacity</li> <li>• costs for driver and driving</li> <li>• date of delivery and opening hours of customers</li> <li>• changes to customer data</li> <li>• special wishes of customers</li> <li>• restrictions to road access</li> <li>• and many more</li> </ul> <p>CATRIN is a computer program that automatically combines customer orders and schedules the delivery tours. It determines optimal assignments between vehicles and delivery tours by taking into account vehicle capacity, times of vehicle availability, and customer wishes.</p>		
<b>Platforms/System Requirements:</b>	Windows 95, Windows 98, Windows NT Intel Pentium 200 MHz, 64 MB RAM, 100 MB free harddrive capacity, 17" color monitor		
<b>Vendor:</b>	alfaplan Management-Software und Consulting GmbH Bismarckring 42 89077 Ulm Germany Phone: +49-(0)731-68435 Fax: +49-(0)731-68419 e-mail: <a href="mailto:catrin@alfaplan.de">catrin@alfaplan.de</a>		
<b>Price:</b>	DM 22,000	<b>Source:</b>	vendor's homepage
<b>Vendor's Homepage:</b>	<a href="http://www.alfaplan.de">http://www.alfaplan.de</a>		
		<b>Last update:</b>	4/1999

**Further Information**

The price indicated above is for the single user standard version. Please contact alfaplan to obtain information about the prices for additional modules, programming and consulting services, digital roadmaps, maintainance contracts, and multi-user licenses.

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>				<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>DynaRoute</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>	DynaRoute is an optimization-based vehicle routing and scheduling system for managing real-time vehicle dispatching. It determines optimal assignments between vehicles and new orders by taking into account vehicle positions, committed order schedules, times of vehicle availability, outstanding order time windows, as well as historical demand profiles in regions of delivery.				
<b>Platforms/System Requirements:</b>	Windows 95, Windows 98, Windows NT				
<b>Vendor:</b>	SAITECH, Inc. 1301 Highway 36 Hazlet, NJ 07733 U.S.A. Phone: (732) 264-4700 Fax: (732) 264-6015 e-mail: <a href="mailto:saitech@monmouth.com">saitech@monmouth.com</a>				
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 12/98	<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.saitech-inc.com">http://www.saitech-inc.com</a>				

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>GeoRoute 5</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	<p>GeoRoute 5 uses a detailed street network database along with specialized routing algorithms to produce efficient point-to-point routes. It can be used daily or periodically to generate routes based on customer orders or regular servicing calendars. In both cases, GeoRoute 5 helps reduce operating costs and better respect servicing commitments.</p> <p>GeoRoute 5 uses a detailed description of the street network, including travel speeds, one-ways and turn restrictions, thus allowing the software to accurately estimate and display all itineraries. The street network data can either be extracted from an in-house geographic information system (GIS) or be acquired from a geographic data vendor. GeoRoute 5 then provides the required tools to update this information.</p> <p>Automatic and interactive tools let you accurately identify customer and depot locations on the street network using the corresponding address. Once this is done, GeoRoute 5 can automatically and accurately calculate travel time and distance between each pair of locations. For each location to be serviced, you can also specify:</p> <ul style="list-style-type: none"> <li>• Acceptable vehicle types.</li> <li>• Types of goods or services to be provided.</li> <li>• Quantity to be pick-up or delivered.</li> <li>• Servicing time, which can be based on quantity.</li> <li>• Possible servicing time windows.</li> </ul>		
<b>Platforms/ System Requirements:</b>	Windows 95, Windows NT, IBM OS/2		
<b>Vendor:</b>	GIRO Enterprises, Inc. 75 Port-Royal East Montreal, H3L3T1 Canada Phone: (514) 383-0404 Fax: (514) 383-4871 e-mail: <a href="mailto:info@giro.ca">info@giro.ca</a>		
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today
<b>Last update:</b>	4/1999		
<b>Vendor's Homepage:</b>	<a href="http://www.giro.ca">http://www.giro.ca</a>		

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>GeoRoute-Postal</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	<p>The new GeoRoute-Postal product provides postal organizations with additional functions and flexibility. These include the ability to plan and optimize mail delivery as well as motorized operations such as mailbox collection, relay bundle delivery, parcel delivery as well as courier services. GeoRoute-Postal map-based displays provide great accuracy in defining locations to be serviced. These can be specific points such as large mailers, mailboxes, or parcel destinations. They can also be one or more street segments, such as in the case of a postal code. Routes can then be built to include either or both types of locations.</p> <p>Similar flexibility is provided in defining available staff resources and travel modes (on foot, bicycle, car, etc.). For each mode, you can specify travel speeds and whether turn restrictions and one-ways must be respected. Modes can also be combined, such as required for "park and loop" servicing, where delivery is done on foot, but a vehicle used to move from one area to the other. Using bus stop locations and schedule information, GeoRoute-Postal can account for travel by public transit between the postal station and route start/end points.</p>		
<b>Platforms/System Requirements:</b>	Windows NT, Novell and TCP/IP networks		
<b>Vendor:</b>	GIRO Enterprises, Inc. 75 Port-Royal East Montreal, H3L3T1 Canada Phone: (514) 383-0404 Fax: (514) 383-4871 e-mail: <a href="mailto:info@giro.ca">info@giro.ca</a>		
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage
<b>Vendor's Homepage:</b>	<a href="http://www.giro.ca">http://www.giro.ca</a>		<b>Last update:</b> 4/1999

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>				<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>GIS/T-Rideshare</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>	GIS/T-Rideshare offers rideshare management tailored to your agency's precise resources and requirements. GIS/T-Rideshare's proven approach is built on the methods and procedures of two established rideshare management systems. To these, it adds the state-of-the-art spatial analysis power of ESRI's popular ArcView desktop Geographic Information System (GIS). Open systems and client/server architecture ensure a solution that fits and grows with your agency's needs.				
<b>Platforms/System Requirements:</b>	Windows 95, Unix				
<b>Vendor:</b>	GIS/Trans, Ltd. 675 Massachusetts Ave. Cambridge, MA 02139 U.S.A. Phone: (617) 354-2771 Fax: (617) 354-8964 e-mail: <a href="mailto:info@gistrans.com">info@gistrans.com</a>				
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 12/98	<b>Last update:</b>	1/1999
<b>Vendor's Homepage:</b>	<a href="http://www.gistrans.com">http://www.gistrans.com</a>				

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>HASTUS</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	<p>The HASTUS transit vehicle and crew scheduling computer system is recognized world-wide for its effectiveness and flexibility. Currently installed in nearly 22 countries in North America, Europe, Africa, and Asia, HASTUS software is a proven time and money saver.</p> <p>The basic HASTUS system provides the tools required for the production of efficient vehicle timetables and operator assignments. Related software offers solutions for passenger information and daily transit operations. HASTUS can efficiently support problems of any size. For example, OC Transpo in Ottawa schedules more than 1200 driver shifts simultaneously, while the Los Angeles MTA schedules approximately 4000 operators working out of 18 depots.</p> <p>The system supports controlled interlining, varied driver relief practices, complex travel time calculations, multiple driver shift types ... and much more!</p>		
<b>Platforms/System Requirements:</b>	Windows NT, Novell and TCP/IP networks		
<b>Vendor:</b>	GIRO Enterprises, Inc. 75 Port-Royal East Montreal, H3L3T1 Canada Phone: (514) 383-0404 Fax: (514) 383-4871 e-mail: <a href="mailto:info@giro.ca">info@giro.ca</a>		
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage
<b>Vendor's Homepage:</b>	<a href="http://www.giro.ca">http://www.giro.ca</a>		<b>Last update:</b> 4/1999

**Further Information**

Pricing is based on property size and problem complexity, so even smaller transit properties can benefit from the power of the HASTUS family of products.

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>					<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>
<b>Komalog</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>	<p>The software-package KOMALOG® is built in a module construction and includes a variety of program parts. This build-up allows the customer to combine such modules especially fitting with his firm.</p> <p>Basic software modules are:</p> <ul style="list-style-type: none"> <li>• freight calculation</li> <li>• groupage business</li> <li>• loading transport</li> <li>• storage, receiving, distribution of goods</li> <li>• freight item management (pallets, transport boxes etc.)</li> <li>• distance information Germany / Europe</li> <li>• fleet costing</li> <li>• managing and supervising dispatch</li> </ul> <p>Special software modules available for:</p> <ul style="list-style-type: none"> <li>• import / export trade</li> <li>• dangerous cargo</li> <li>• optimization</li> <li>• full costing, variable costing</li> <li>• consignment monitoring and tariffs</li> <li>• cost comparisons and current status</li> <li>• distribution costs</li> <li>• heavy load transporting</li> </ul> <p>Obviously the KOMALOG® software is capable of running in a lan and there is no problem with data exchange to branch offices. Unfortunately the KOMALOG® software is only available in German at this time. The development of an international version will be finished until Expo 2000, Hannover.</p>				
<b>Platforms/ System Requirements:</b>	Windows 95, Windows NT				
<b>Vendor:</b>	TRANSDATA Soft- und Hardware GmbH Heidsieker Heide 114 33739 Bielefeld Germany Phone: +49 (0)5206 91 06-0 Fax: +49 (0)5206 91 06-90 e-mail: <a href="mailto:komalog@transdata.net">komalog@transdata.net</a>				
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage	<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.transdata.net">http://www.transdata.net</a>				

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Un">University of Karlsruhe</a>					<a href="http://www.uni-karlsruhe.de/Un">http://w ww.uni- karlsruh e.de/Un i/</a>
<b>Load Manager</b>					
<b>Category:</b>	Commercial	Vehicle Routing			
<b>Key Features:</b>					
<b>Platforms/ System Requirements:</b>					
<b>Vendor:</b>	Roadnet Technologies, Inc. 2311 York Road, Timonium, MD 21093 U.S.A. Phone: (410) 560-4298 Fax: (410) 560-4328 e-mail: <a href="mailto:marketing@roadnet.ups.com">marketing@roadnet.ups.com</a>				
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today	<b>Last update:</b>	5/1997
<b>Vendor's Homepage:</b>	<a href="http://www.roadnet.com">http://www.roadnet.com</a>				

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>Prophesy Mileage 7.1</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	Prophesy Mileage 7.1 is a fully integrated Windows-based mileage and routing decision support system. It routes to hundreds of thousands of cities, towns, zip codes, intersections, truck stops, and user-defined points quickly and easily. Prophesy Mileage 7.1 can calculate both short-distance rating mileages and detailed practical routings, depending upon the needs of your company's transportation operation. Thousands of shippers, carriers, brokers, and other transportation professionals utilize the ease and power of Prophesy Mileage 7.1 every day.		
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT		
<b>Vendor:</b>	Prophesy Transportation Software, Inc. 204-C West Newberry Road Bloomfield, CT 06002 U.S.A. Phone: (800) 776-6706 Fax: (860) 243-2619 e-mail: <a href="mailto:info@mile.com">info@mile.com</a>		
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage
<b>Vendor's Homepage:</b>	<a href="http://www.mile.com/mileage.html">http://www.mile.com/mileage.html</a>		
		<b>Last update:</b>	4/1999

## Further Information

Prophesy offers a [30 day trial](#) of Mileage 7.1. If, for any reason, within 30 days of purchase you decide that you're not completely satisfied with Prophesy Mileage 7.1 for Windows you may return the software for a complete refund (less shipping & handling).

### Prophesy Mileage 7.1 Features

- Provides mileage summary by state for fuel tax reporting.
- Short distance or practical mileages.
- Hundreds of thousands of driving locations.
- Cities, zip codes, highway junctions, and truckstops all on-line.
- Optimize routes by quickest driving time or shortest distance.
- Set vehicle heights and other commercial restrictions.
- No transaction fees or lookup restrictions.
- Optional integration with other Prophesy for Windows transportation software.
- Complete routing reports with route and exit numbers.
- Easy installation and on-line help.
- Calculates total driver miles and times for any trip.
- Includes toll road and seasonal road closure avoidance features.
- Route to user-defined latitude/longitude coordinates.

### Prophesy Mileage 7.1 Versions

- **Prophesy ShortCut 7.1** provides industry standard commercial rating mileage lookups between over 1,000,000 North American driving locations. It allows mileages to be obtained right on your own computer, unlike clumsy and costly transaction-based mileage systems. And, there is never a limit to the number of transactions or mileage lookups. It allows automatic optimization and re-sequencing of points in a route, and the ability to adjust mileages by set percentages, or to store pre-defined miles for any point pairs. ShortCut is registered as a legal tariff source

with the Surface Transportation Board (formerly ICC) ICC#ISIC-100 and provides an accepted source of industry-standard rating miles.

- **Prophesy ProDriver 7.1** provides practical driving routes and mileages over non-restricted commercial roads. ProDriver will route vehicles between virtually any North American driving locations. ProDriver reports include driving directions, estimated driving time, costs, and mileage summaries by state (an excellent tool for fuel-tax reporting) for any trip. It is an ideal decision support system for logistics management, driver directions, payment calculation, trip costing analysis, and rating/quotation of services. ProDriver is registered as a legal tariff source with the Surface Transportation Board (formerly ICC) ICC#ISIC-101 and is currently used by several states as their standard tool for fuel tax auditing.
  - **Prophesy Plus 7.1** combines all the features of ShortCut and ProDriver into one easy-to-use system. It's ideal for the company that needs both industry standard rating mileages and practical commercial driving mileages, and is fully accepted by the Surface Transportation Board (formerly ICC) as an accurate source of legal tariff mileages. Like ShortCut and ProDriver, Plus allows routing to virtually all North American cities, zip codes, U.S. Postal facilities, military bases, airports, truck stops, lat./long. coordinates, highway intersections and user-defined points. Also, users have great control over how routes are calculated since they may specify attributes such as vehicle height, toll road usage, seasonal road conditions, commercial vehicle restrictions, and speed adjustments to customize the route generated based upon your own fleet requirements.
-

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>OVERS</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	Can optimize routes across multiple time periods, respect space/time constraints, optimize number and location of terminals and service areas.		
<b>Platforms/ System Requirements:</b>	PC		
<b>Vendor:</b>	Bender Management Consultants 1755 Jefferson Davis Highway Arlington, VA 22202 U.S.A. Phone: (703) 412-0840 Fax: (703) 412-0833 e-mail: <a href="mailto:bendermac@interramp.com">bendermac@interramp.com</a>		
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 6/97
		<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>			

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Un">University of Karlsruhe</a>					<a href="http://www.uni-karlsruhe.de/Un">http://w ww.uni- karlsruh e.de/Un i/</a>
<b>ROADNET 5000</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>					
<b>Platforms/ System Requirements:</b>					
<b>Vendor:</b>	Roadnet Technologies, Inc. 2311 York Road Timonium, MD 21093 U.S.A. Phone: (410) 560-4298 Fax: (410) 560-4328 e-mail: <a href="mailto:marketing@roadnet.ups.com">marketing@roadnet.ups.com</a>				
<b>Price:</b>		<b>Source:</b>	OR/MS Today	<b>Last update:</b>	5/1997
<b>Vendor's Homepage:</b>	<a href="http://www.roadnet.com">http://www.roadnet.com</a>				

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>					<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>
<b>RouteSmart</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>	<p>RouteSmart evaluates the numerous ways in which the driver (or drivers) could travel over the streets and service customers on their route while factoring for the geographic distribution of stops against a street network and other operational constraints (customer service time restrictions, vehicle capacities, one-way streets, etc.). The RouteSmart system pays close attention to details and creates optimized dispatch plans which minimize total operating costs for servicing a geographic area.</p> <p>RouteSmart enables route planners with a powerful set of tools for evaluating different routing plans which are developed with the street network, geographic distribution of customers, and operational constraints of the industry in mind. What's best, RouteSmart visualizes the data route planners have to sort through to give them a map-based representation of the stop locations, street network, routes and sequential ordering of customers for service.</p>				
<b>Platforms/ System Requirements:</b>	Windows				
<b>Vendor:</b>	RouteSmart Technologies 8850 Stanford Blvd., Suite 2600 Columbia, MD 21045 U.S.A. Phone: (800) 977-7284 Fax: (410) 290-0334 e-mail: <a href="mailto:info@routesmart.com">info@routesmart.com</a>				
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 6/97	<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.routesmart.com">http://www.routesmart.com</a>				

## Further Information

### Why Choose RouteSmart ?

- RouteSmart is a comprehensive route optimization software system developed by a company whose primary focus and expertise is on routing solutions.
- RouteSmart has an extensive client base and experience in software development with a keen understanding of the routing needs in the industries we serve.
- RouteSmart provides the only turnkey routing solution utilizing GIS technology. RouteSmart is a scalable routing solution that can be delivered on a desktop PC, workstation or via the internet.
- RouteSmart is provided direct to the end user. This allows for a close, collaborative relationship with our customers.
- RouteSmart is a precision routing and scheduling product designed to model and account for factors of real-world, vehicle routing logistics. RouteSmart is used by companies around the world with some of the most demanding routing needs.
- RouteSmart promotes dramatic cost savings to help companies increase profitability and provide better customer service.

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>				
<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>				
<b>SHIPCONS II</b>				
<b>Category:</b>	Commercial		Vehicle Routing	
<b>Key Features:</b>	<p>SHIPCONS II is a management support system (MSS) designed to assist transportation planners and dispatchers to make better decisions about transportation resources. SHIPCONS II combines the expert knowledge of the company's dispatchers, the mass of data about customers, orders, trucks, and distribution centers, and the power of the most advanced optimization technology available to build and dispatch optimal loads of packaged goods to customers.</p> <p>Working through an easily understood graphical user interface, the dispatcher can evaluate various shipping options to select that shipment pattern which best balances customer service requirements with lowest operating cost.</p>			
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT Pentium processor, 32MB RAM, 1GB HD, 17-inch monitor			
<b>Vendor:</b>	Insight, Inc. 19820 Village Office Court Bend, OG 97702 U.S.A. Phone: (541) 388-6998 Fax: (541) 388-9884 e-mail: <a href="mailto:dbausch@insight-mss.com">dbausch@insight-mss.com</a>			
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 6/97	<b>Last update:</b> 4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.insight-mss.com">http://www.insight-mss.com</a>			

### Further Information

**How does SHIPCONS II work ?**

SHIPCONS II receives data about the orders to be shipped, the vehicles available, and policy factors to be considered in building and dispatching shipments. The dispatcher can easily and quickly change any of these data through a graphical user interface. The raw data are processed with the rules and policies from the user's expert knowledge base to generate a large number of feasible delivery routes and schedules. All of these feasible routes are then fed into the optimizer, a mixed integer linear program, which selects that set of routes and schedules which assures all orders are delivered at lowest cost. This optimal dispatch is then conveyed back to the dispatcher through a graphical user interface for implementation or further adjustment as the dispatcher determines appropriate.

**The types of input data required by SHIPCONS II include:**

- Source data - origin locations of plants and DCs
- Product data - size and weight characteristics
- Customer data - locations and special considerations
- Order data - quantity ordered by product by customer
- Vehicle data - capacities, costs, and locations
- Work standards - shift hours, overtime rules, etc.
- Geographic network - distances, drive times, etc.

**The following shipment alternatives can be considered in finding the best overall pattern:**

- Selective early shipment to obtain full truckloads
- Local delivery - owned trucks
- Local delivery - hired trucks
- Overnight delivery - owned trucks

- Packaged goods carriers (UPS, etc.)
- LTL
- TL
- TL with stopoffs
- TL to pool point with local delivery beyond (trailer switch)
- TL to pool point with LTL beyond (stopoffs before/after pool)
- Outbound with backhauls

**Choosing among these alternatives is subject to a series of policies and constraints which can be imposed by the dispatcher or by the system automatically:**

- Truck/trailer weight and cube capacity limitations
- Carrier limits on stopoffs
- Interstate and intrastate restrictions
- Special equipment (liftgates, refrigeration, etc.)
- Truck operating radius around ship-from location
- Minimum capacity available before backhauls considered
- Delivery time windows.

**For each scenario executed the user receives a set of outputs in either graphical or hard copy form. These include:**

- List of orders with optimal shipment assignment
- Summary of shipments by source
- Cost summaries by shipment mode
- Truck summary showing stops, hours, and distance for each vehicle.

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Un">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Un">http://w ww.uni- karlsruh e.de/Un i/</a>	
<b>Territory Planner</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>			
<b>Platforms/ System Requirements:</b>	Windows 95, Windows NT, IBM OS/2		
<b>Vendor:</b>	Roadnet Technologies, Inc. 2311 York Road, Timonium, MD 21093 U.S.A. Phone: (410) 560-4298 Fax: (410) 560-4328 e-mail: <a href="mailto:marketing@roadnet.ups.com">marketing@roadnet.ups.com</a>		
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today <b>Last update:</b> 5/1997
<b>Vendor's Homepage:</b>	<a href="http://www.roadnet.com">http://www.roadnet.com</a>		

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe/">http://www.uni-karlsruhe.de/University_of_Karlsruhe/</a>	
<b>TransCAD</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	<p>TransCAD is the first and only Geographic Information System (GIS) designed specifically for use by transportation professionals to store, display, manage, and analyze transportation data. TransCAD combines GIS and transportation modeling capabilities in a single integrated platform, providing capabilities that are unmatched by any other package. TransCAD can be used for all modes of transportation, at any scale or level of detail. TransCAD provides:</p> <ul style="list-style-type: none"> <li>• A powerful GIS engine with special extensions for transportation</li> <li>• Mapping and visualization tools designed for transportation applications</li> <li>• Application modules for routing, travel demand forecasting, and location modeling</li> </ul>		
<b>Platforms/System Requirements:</b>	Windows 95, Windows 98, Windows NT		
<b>Vendor:</b>	<p>Caliper Corporation  1172 Beacon St.  Newton, MA 02461-9926  U.S.A.  Phone: (617) 527-4700  Fax: (617) 527-5113  e-mail : <a href="mailto:info@caliper.com">info@caliper.com</a></p>		
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 6/97
		<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.caliper.com">http://www.caliper.com</a>		

## Further Information

Caliper has just completed an updated version of TransCAD, Version 3.5. TransCAD 3.5 is intended for use exclusively under the Microsoft Windows operating environment, and runs under Windows 95, Windows 98, and Windows NT. TransCAD 3.5 has many significant enhancements. These include support for long filenames, coded fields, and import of Excel files, greatly expanded geographic editing capabilities, a better interface for joining tables to geography, improved ODBC functionality and performance, and easier programmability.

Some of the improvements, such as faster access to sparse matrices, come from changes in file formats for line layers and matrices. There are utilities to convert version 3.x files to 3.5, but there is no backward compatibility so files created with version 3.5 can only be used with the new version.

To obtain more information about the new features of TransCAD, Version 3.5, click [here](#).

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe">University of Karlsruhe</a>					<a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe/">http://www.uni-karlsruhe.de/University_of_Karlsruhe/</a>
<b>Trapeze-FX</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>	FX is a fixed route schedule builder and vehicle and driver assignment system. Its comprehensive functionality and sophisticated algorithms generate efficient solutions that recognize savings in both labor and vehicle operating costs. FX has been specifically designed for full automation of the scheduling process while enabling manual intervention for tailor-made solutions.				
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT				
<b>Vendor:</b>	Trapeze Software Group Greenway-Hayden Loop Building A, Suite 200 Scottsdale, AZ 85260 U.S.A. Phone: (602) 627-8400 Fax: (602) 627-8411 e-mail: <a href="mailto:info@mail.trapezesoftware.com">info@mail.trapezesoftware.com</a>				
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 6/97	<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.trapezesoftware.com">www.trapezesoftware.com</a>				

### Further Information

**Trapeze-FX's functionality:**

**Route Definition:**

- Define patterns or route variations at time point, bus stop or street level
- Define and geocode time points and bus stops
- Plot new or altered transit routes

**Schedule/Trip Building:**

- Build timetables at any frequency for all modes of transit which can easily be adjusted for detours or calendar variations
- Determine connections and transfers in timetables
- Interpolate arrival times at all bus stops within the service area
- Utilize running-time templates to support the various service levels

**Blocking/Vehicle Assignment:**

- Choose manual, optimal or heuristic blocking methods
- Interline routes or trips minimizing vehicle use while respecting predetermined union rules
- Assign each vehicle block to a garage, optimally minimizing the deadhead distance and respecting garage limitations
- Calculate deadhead times, distances, and suggested deadhead routes

**Runcutting/Driver Assignment:**

- Minimize cost while adhering to all aspects of union agreements
- Respect both 'hard' and 'soft' rules when assigning drivers
- Ceate cost effective yet 'streetable' solutions
- Calculate travel times based on time of day or type of scheduled service
- Generate driver turn-lists with the click of a button

**Rostering/Weekly Crewing:**

- Support various types of crewing rules
- Create both 4-day and 5-day rosters, with or without consecutive days off
- Modify costing routines to calculate and minimize weekly makeup and overtime

**The Malteze Database**

All modules in the TRAPEZE® system access data through the Malteze database enabling data to be shared seamlessly amongst the various transit modes (flexible, fixed route, paratransit etc.). Malteze streamlines the processes of planning, scheduling, customer information and operations by eliminating the export and translation of data.

**Technology Standards**

The TRAPEZE® system is a true 32-bit software application built for Windows 95/NT. Features include:

- Three-tier client/server architecture
  - One common graphical interface
  - Embedded geographical information systems (GIS) functionality
  - TCPIP, IPX/SPX and named pipes protocols
  - ODBC and SQL compliant database management systems
  - Configurable for local and wide area networks
  - Enabled for the Internet
  - Configurable for single-user and multi-user environments
  - Standard and ad hoc reporting
  - Enabled for other technologies (automatic vehicle location and computerized voice)
-

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe">University of Karlsruhe</a>				<a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe/">http://www.uni-karlsruhe.de/University_of_Karlsruhe/</a>	
<b>Trapeze-Taxi</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>	<p>TRAPEZE@TAXI is a computer-assisted voice dispatching, routing and management application that is actually designed by taxi companies for taxi companies.</p> <p>With an extensive range of options and functions, you have the flexibility to tailor the system to the way you run your operation and effectively organize your fleet whether you have 10 to over 1000 or more vehicles.</p> <p>With TRAPEZE@TAXI you can:</p> <ul style="list-style-type: none"> <li>• Instantly dispatch cars and notify drivers of changes</li> <li>• Reduce errors and increase efficiency with the telephone Caller ID feature</li> <li>• Monitor live dispatches and view trip histories</li> <li>• Build a customer database – who they are, where they're going and how often</li> <li>• Easily generate reports such as operation efficiency over a five-hour or even a five-year period</li> </ul>				
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT				
<b>Vendor:</b>	<p>Trapeze Software Group                  Greenway-Hayden Loop                  Building A, Suite 200                  Scottsdale, AZ 85260                  U.S.A.                  Phone: (602) 627-8400                  Fax: (602) 627-8411                  e-mail: <a href="mailto:info@mail.trapezesoftware.com">info@mail.trapezesoftware.com</a></p>				
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage	<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.trapezesoftware.com">www.trapezesoftware.com</a>				

### Further Information

**Trapeze-Taxi's functionality:**

**Call Management:**

- TAPI automatic telephone Caller ID
- Notify drivers of changes instantly
- Record special requirements
- Callbacks are recognized
- Filter out unwanted callers

**Dispatch Management:**

- Assign rides instantly
- Accurate dispatch information to driver
- Unlimited frequencies, zones and subs
- Automatic sorting of streets and zones
- Multiple views of ride/vehicle status
- Color-coded "Quick-Vu" of vehicle status
- History Database gives quick access to information

**Customer Management:**

- Identify your frequent customers
- Determine peak times and zones

**Prospectives:**

TRAPEZE®TAXI easily adapts to MDT (Mobile Data Terminals) and AVL (Automatic Vehicle Location) technology. TRAPEZE®TAXI can integrate with other Trapeze products such as TRAPEZE®PASS, an application deployed in over 200 paratransit operations. Taxi companies can now easily extend business interests to the public transportation, paratransit, and welfare-to-work transportation sectors.

Our integrated system allows you to scale up your operation quickly and add more technology components as you go:

- Mapping
- Cashiering System
- Debit/Credit Direct Authorization
- Mobile Data Terminals / Global Positioning Systems

**Technology Standards**

The TRAPEZE® system is a true 32-bit software application built for Windows 95/NT. Features include:

- Three-tier client/server architecture
  - One common graphical interface
  - Embedded geographical information systems (GIS) functionality
  - TCPIP, IPX/SPX and named pipes protocols
  - ODBC and SQL compliant database management systems
  - Configurable for local and wide area networks
  - Enabled for the Internet
  - Configurable for single-user and multi-user environments
  - Standard and ad hoc reporting
  - Enabled for other technologies (automatic vehicle location and computerized voice)
-

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>				
<b>Wintour</b>				
<b>Category:</b>	Commercial	Vehicle Routing		
<b>Key Features:</b>				
<b>Platforms/ System Requirements:</b>				
<b>Vendor:</b>	Soloplan GmbH Software für Logistik und Planung Cambodunumweg 11 D-87437 Kempten phone : 0831/5 74 07-0 fax : 0831/5 74 07-20 email : <a href="mailto:soloplan@online-service.de">soloplan@online-service.de</a>			
<b>Price</b>	call	<b>Source:</b>		<b>Last update:</b> 12/1997
<b>Vendor's Homepage:</b>	<a href="http://www.soloplan.de/dtwint.htm">http://www.soloplan.de/dtwint.htm</a>			

### Further Information

Demo available at vendors homepage.

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>Direct Route</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	DirectRoute is designed to help load planners create the most efficient routes possible. Its powerful routing engine designs routes based on the customer location, vehicle capacity, and customer time windows, thus helping you to ensure maximum truck usage, minimum miles and timely service.		
<b>Platforms / System Requirements:</b>	DOS, Windows 95, Windows NT		
<b>Vendor:</b>	Appian Logistics Software 1601 Greenbriar Place, Suite J Oklahoma City, OK 73159 U.S.A. Phone: (800) 893-1250 Fax: (405) 692-1698 e-mail: <a href="mailto:appian@ionet.net">appian@ionet.net</a>		
<b>Price:</b>	\$12,500	<b>Source:</b>	OR/MS Today
		<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.appianlogistics.com">http://www.appianlogistics.com</a>		

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>					
<b>Easy Router</b>					
<b>Category:</b>	Commercial	Vehicle Routing			
<b>Key Features:</b>	<p>Easy Router is the first ever Internet-based routing service. With Easy Router your field service employees, your truck drivers, your sales personnel, or anyone who spends most of their workday getting in and out of a car or truck will be contributing to your bottom line. They will no longer be runaway costs and a nightmare to manage.</p> <p>Easy Router accounts for:</p> <ul style="list-style-type: none"> <li>• Actual Road Distance</li> <li>• Turn Restrictions and One-way Streets</li> <li>• Vehicle Capacities</li> <li>• Driver Capabilities</li> <li>• Appointment Time Windows.</li> <li>• Insures that your data is password protected and secure.</li> </ul>				
<b>Platforms/ System Requirements:</b>	Windows				
<b>Vendor:</b>	Lightstone Group, Inc. 250 Old Country Road Mineola, NY 11501 U.S.A. Phone: (516) 294-7505 Fax: (516) 294-5543 e-mail: <a href="mailto:lsginfo@lightstone.com">lsginfo@lightstone.com</a>				
<b>Price:</b>	You pay for the routes as you generate them.	<b>Source:</b>	vendor's homepage	<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.lightstone.com">http://www.lightstone.com</a>				

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>GeoRoute-Municipal</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	<p>GeoRoute-Municipal is a powerful map-based routing software that is specifically designed for the automatic creation of routes for city services such as:</p> <ul style="list-style-type: none"> <li>• Residential and commercial waste collection</li> <li>• Recycling collection</li> <li>• Street cleaning</li> <li>• Salt and sand spreading</li> <li>• Snow removal</li> </ul> <p>The software provides all the required tools to prepare the data, create and optimize the routes, and output the results in the form of maps or reports. Resulting productivity gains typically range from 5 to 15%.</p> <p>GeoRoute-Municipal's map-based displays provide great accuracy in defining locations to be serviced. These can be specific points such as buildings in the case of commercial waste collection. They can also be entire segments, such as in the case of residential waste collection. Routes can then be built to include either or both types of locations.</p> <p>GeoRoute-Municipal uses a detailed description of the street network, including travel speeds, one-ways and turn restrictions, thus allowing the software to accurately estimate and display all itineraries. The street network data can either be extracted from an in-house geographic information system (GIS) or be acquired from a geographic data vendor. GeoRoute-Municipal then provides the required tools to update this information.</p>		
<b>Platforms/System Requirements:</b>	Windows NT, Novell and TCP/IP networks		
<b>Vendor:</b>	GIRO Enterprises, Inc. 75 Port-Royal East Montreal, H3L3T1 Canada Phone: (514) 383-0404 Fax: (514) 383-4871 e-mail: <a href="mailto:info@giro.ca">info@giro.ca</a>		
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage
<b>Vendor's Homepage:</b>	<a href="http://www.giro.ca">http://www.giro.ca</a>		<b>Last update:</b> 4/1999

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>GIRO/ACCESS</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	<p>Transit for the disabled reservation and scheduling: The demand for specialized or custom transit services for the disabled, the elderly, and other groups has been growing rapidly. With this expansion has come the need to reduce the cost per trip while increasing availability of rides. GIRO/ACCES is designed to meet the needs of large paratransit services with centralized call-taking and scheduling, as well as multiple service providers.</p> <p>GIRO/ACCES permits easy access to up-to-date customer information and automates the production of the numerous reports required by management and government. The system furnishes rapidly and accurately, all necessary information on client trip requests, in order to respond to queries from drivers, office staff, and customers. It provides fully automated scheduling tools that build runs that optimize productivity while respecting service policies.</p>		
<b>Platforms/System Requirements:</b>	Windows		
<b>Vendor:</b>	GIRO Enterprises, Inc. 75 Port-Royal East Montreal, H3L3T1 Canada Phone: (514) 383-0404 Fax: (514) 383-4871 e-mail: <a href="mailto:info@giro.ca">info@giro.ca</a>		
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage
<b>Vendor's Homepage:</b>	<a href="http://www.giro.ca">http://www.giro.ca</a>		<b>Last update:</b> 4/1999

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>				
<b>GIS/T-SchoolBus</b>				
<b>Category:</b>	Commercial	Vehicle Routing		
<b>Key Features:</b>	GIS/T-SchoolBus manages the scheduling and assignment of students to school bus routes.			
<b>Platforms/ System Requirements:</b>	Windows 95, Unix			
<b>Vendor:</b>	GIS/Trans, Ltd. 675 Massachusetts Ave. Cambridge, MA 02139 U.S.A. Phone: (617) 354-2771 Fax: (617) 354-8964 e-mail: <a href="mailto:info@gistrans.com">info@gistrans.com</a>			
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 12/98	<b>Last update:</b> 1/1999
<b>Vendor's Homepage:</b>	<a href="http://www.gistrans.com">http://www.gistrans.com</a>			

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe">University of Karlsruhe</a>					<a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe/">http://www.uni-karlsruhe.de/University_of_Karlsruhe/</a>
<b>Intertour</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>	<p>INTERTOUR, the PC-planning-program for all logistics managers, vehicle managers and disponents enables its users to find the best solution for complex planning programs. INTERTOUR helps you to reduce costs effectively. INTERTOUR can be used to:</p> <ul style="list-style-type: none"> <li>• do the daily planning of the operative tour-planning with changing points of departure</li> <li>• plan Europe-wide</li> <li>• optimize inner-city tours with INTERTOUR/City</li> <li>• plan kombined collection and distribution tours</li> <li>• and much more !</li> </ul>				
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT				
<b>Vendor:</b>	PTV GmbH Stumpfstr. 1 76131 Karlsruhe Germany Phone: +49 (0)721 9651-500 Fax: +49 (0)721 9651-599 e-mail: <a href="mailto:ptv@ptv.de">ptv@ptv.de</a>				
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage	<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.ptv.de">http://www.ptv.de</a>				

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>LoadExpress Plus</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	<p>LoadExpress Plus for Windows is Prophecy Transportation Software's newest vehicle optimization and scheduling technology. It bridges the gap between manual load planning methods and complex automated loadbuilding systems. The only load planning and route scheduling system of its kind, LoadExpress Plus offers a simple, powerful, and flexible choice for building and optimizing routes, scheduling deliveries, and analyzing distribution patterns.</p> <p>LoadExpress Plus supports logistics management considerations such as stop order definition, delivery time windows, time and distance optimization, variable fleet size, DOT regulations, and much more. It links with existing order entry, dispatch, and other transportation systems, and is flexible enough to meet both straightforward and complex distribution requirements.</p>		
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT		
<b>Vendor:</b>	Prophecy Transportation Software, Inc. 204-C West Newberry Road Bloomfield, CT 06002 U.S.A. Phone: (800) 776-6706 Fax: (860) 243-2619 e-mail: <a href="mailto:info@mile.com">info@mile.com</a>		
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today
		<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.mile.com/loadexpress.html">http://www.mile.com/loadexpress.html</a>		

## Further Information

You can download a [demo version of LoadExpress Plus](#).

### LoadExpress Plus Features

- Optional automatic loadbuilding or interactive load planning and assembly.
- Full-color Windows-based mapping interface for state-of-the-art power and ease of use.
- Instant access to all summary and detail data for loads, trucks, and routes.
- Dynamic on-screen display of used and available vehicle capacities.
- Drag and drop editing makes transferring orders between loads a breeze.
- Integrates with Prophecy Mileage database for easy optimization of delivery sequences.
- Easy integration with other applications such as order entry systems, dispatch software, on-board computers, and Global Positioning Systems (GPS).
- Complete support as part of the LoadExpress Plus package.

### LoadExpress Plus Benefits

- Enables better utilization of fleet resources.
- Automatically optimizes your resources in accordance with all constraints and regulations.
- Allows dynamic recalculation of scheduling based on late-breaking information.
- Frees up the workload of dispatchers and load planners so they can concentrate on customer service.
- Can handle unlimited numbers of depots and delivery locations.
- Integrates with and complements existing computerized systems.
- Allows interactive analysis and decision making right on your PC.

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>Manugistic Routing &amp; Scheduling</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	<p>Transportation Management dramatically cuts supply chain cost and order cycle time. It provides global transportation planning and execution capabilities, whether you are a manufacturer, distributor, retailer, or third-party service provider. Your transportation professionals worldwide can reduce freight costs and access real-time shipment information.</p> <p>Transportation Management's global optimization and execution gives you the capability of running your entire transportation organization with a single application. Its multi-modal capabilities help you orchestrate the most efficient combinations of modes to fully optimize your transportation plans. Its optimization creates continuous moves, allowing you to maximize carrier equipment and reduce costs. Its execution capabilities allow you to pay your freight bills, track your shipments, and analyze and report on historical transportation data.</p>		
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT		
<b>Vendor:</b>	<p>Manugistics, Inc.                  2115 E. Jefferson St.                  Rockville, MD 20852                  U.S.A.                  Phone: (301) 984-5000                  Fax: (301) 984-5370                  For more information complete the <a href="#">Information Request Form</a>.</p>		
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 6/97
		<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.manugistics.com">http://www.manugistics.com</a>		

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>Optrak</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	Optrak cuts distribution costs by planning efficient loads and vehicle routes, making the best use of your resources. Sophisticated optimisation delivers better solutions over a wide range of problems.		
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT Pentium II processor, 64 MB RAM, 4 GB hard disk, 17"+ high-resolution monitor (1024x768 or better)		
<b>Vendor:</b>	Optrak Distribution Software Ltd Cawthorne House 51 St. Andrew Street Hertford Hertfordshire SG14 1HZ England Phone: 01992 411 000 Fax: 01992 411 001 e-mail : <a href="mailto:sales@optrak.co.uk">sales@optrak.co.uk</a>		
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage
<b>Vendor's Homepage:</b>	<a href="http://www.optrak.co.uk">http://www.optrak.co.uk</a>		<b>Last update:</b> 4/1999

<b>Plantour</b>				
<b>Category:</b>	Commercial	Vehicle Routing		
<b>Key Features:</b>				
<b>Platforms/ System Requirements:</b>	Windows 95, Unix			
<b>Vendor:</b>	PLS Unternehmensberatung GmbH Stuttgart Zettachring 12 70567 Stuttgart Germany Phone: +49 - 711 -720270 Fax: +49 - 711 - 7202720			
<b>Price:</b>		<b>Source:</b>		<b>Last update:</b> 5/1997
<b>Vendor's Homepage:</b>				

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>				
<b>RiMMS</b>				
<b>Category:</b>	Commercial	Vehicle Routing		
<b>Key Features:</b>	<p>RiMMS routing and scheduling software solutions enable businesses, from a broad range of industries, to better manage their resources in the field by improving customer service, increasing productivity and efficiency, and ultimately reducing overall operational costs. RiMMS helps users to cost-effectively custom-design route delivery schedules and assign tasks to resources. The system integrates complex mapping technology to rapidly create full-color maps with precise text route instructions, based on various parameters from the users database.</p> <p>RiMMS' ability to make system updates in real-time allows companies to be more responsive to their customers by permitting last-minute changes to routes and schedules.</p>			
<b>Platforms/ System Requirements:</b>	Windows			
<b>Vendor:</b>	Lightstone Group, Inc. 250 Old Country Road Mineola, NY 11501 U.S.A. Phone: (516) 294-7505 Fax: (516) 294-5543 e-mail: <a href="mailto:lsginfo@lightstone.com">lsginfo@lightstone.com</a>			
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 6/97	<b>Last update:</b> 4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.lightstone.com">http://www.lightstone.com</a>			

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>					<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>
<b>RoutePro Dispatcher</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>	<p>RoutePro Dispatcher includes the most powerful and extensive routing and scheduling optimization features in the industry for companies that use private or dedicated fleets. With it, you can either build new routes from scratch, or adapt daily fluctuations to current master routes. Then dispatch your fleet, monitor and react to any changes, track and measure performance, and capture information for strategic route planning. RoutePro Dispatcher optimizes even the largest and most complex routing situations, thanks to its workstation class performance and efficient handling of enterprise scale data.</p> <p>RoutePro Dispatcher features the industry's best interactive map-based graphics, making it easy to understand and change routing strategies. These advanced graphics illustrate costly or underutilized routes, and extensive drag and drop capabilities allow you to edit route assignments and sequences. Simply drag and drop to:</p> <ul style="list-style-type: none"> <li>• assign orders and customers to routes</li> <li>• move orders and customers between routes</li> <li>• rearrange route sequences</li> <li>• "lasso" groups of customers or orders to add to a route</li> </ul>				
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT				
<b>Vendor:</b>	CAPS Logistics, Inc. 2700 Cumberland Parkway Atlanta, GA 30339-3369 U.S.A. Phone: (770) 432-9955 Fax: (770) 438-9630 e-mail: <a href="mailto:info@caps.com">info@caps.com</a>				
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 6/97	<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.caps.com">http://www.caps.com</a>				

### Further Information

**Key Features include:**

**Operationally Build Routes**

- Easily attach to other systems for download and upload
- Quickly build routes for pending service
- Adapt master routes from RoutePro Designer™ to actual orders
- Use territory assignments from RoutePro Designer
- Edit routes easily to maximize utilization

**Dynamically Dispatch Routes**

- Update orders dynamically with timer
- Watch automatic updates of route statuses
- Use GPS information to track vehicle and driver locations
- Dynamically adjust remaining portions of routes based on new orders
- Integrate easily with any mobile data or GPS system

**Evaluate Routes**

- Automatically keep history of all route plans
- Enter actuals or integrate with any route-recording system
- Print exception reports comparing actual vs. planned
- Validate stop time and drive time estimations

#### **Technology**

- Unlimited number of depots, sites, orders, vehicles, and routes
- Unlimited number of time windows and appointments
- Set restrictions based on business rules
- Easy to use drag and drop route editing
- Route over multiple days with layover rules
- Route inbound and outbound simultaneously
- Use domiciled drivers and cross-docking to minimize driver layovers and maximize equipment utilization
- Windows™ 95 style point and click interface

#### **Time Windows**

- All routing functions obey time windows
- Any number of time windows per site
- Specify weekday and weekend patterns
- Use flexible time windows to allow early or late service
- Manually override windows when needed
- Maximum allowable wait times
- Separate time windows for order pickup and order delivery
- Override time windows with order appointments .

#### **Orders**

- Define custom order types (delivery, urgent priority, inter-company, etc.)
- Specify pickup site and delivery site
- Indicate product type for each order
- Specify quantity units by product type
- Specify how to convert product quantities into standard vehicle capacities
- Define up to three standard vehicle quantities
- Specify order priorities
- Specify vehicle sharing restrictions between product types/ order types
- Create planning orders based on historical order information

#### **Work Zones and Territories**

- Interactively or automatically create work zones
- Create territories containing sets of work zones
- Partition customers by work zone
- Restrict routes and drivers by work zone or territory
- Build routes to work zones at an aggregate level
- Balance work in work zones and territories
- Create master zone routes servicing work zones

#### **Easy-to-Use Interface**

- Control dispatcher interface
- Pre-setup batch processes for push-button dispatcher functions

#### **Sites**

- Unlimited number of sites
- Define custom site types (customers, plants, distribution centers, preferred customers, etc.)
- Include address, contact, and phone information
- Specify dock types or other site characteristics to control vehicle restrictions

#### **Vehicles**

- Define custom vehicle types (28', 48', dedicated, side-loader, etc.)
- Can specify unique origin for each vehicle
- Can specify unique final destination for each vehicle
- Automatically determine origin and destination based on route
- Multiple trips and multiple routes
- Specify multiple vehicle types with unique characteristics

- Costs:
  - Fixed
  - Minimum
  - Cost per distance
  - Cost per hour
  - Stop-off
- Specify starting time of vehicle or find best starting time
- Vehicle to site restrictions based on dock type, site type, or territory
- Vehicle to order restrictions based on order type or product type
- Enforce last-in, first-out loading of orders
- Vehicle assignment Gantt charts for visual vehicle-route allocation

**Drivers and Crews**

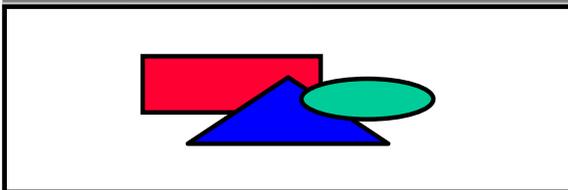
- Set up teams of any number of drivers
  - Restrict drivers by vehicles, work zones, and territories
  - Costs by driver and team
  - Driver breaks by driver or team
  - Labor restrictions
  - Driver Assignment Gantt charts for visual driver-route allocation
-

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>				
<b>Routronics 2000</b>				
<b>Category:</b>	Commercial	Vehicle Routing		
<b>Key Features:</b>	<p>The ROUSTRONIC 2000 system allows foremen to plan delivery routes hours in advance of actual freight arrival, giving the "best route" and proper delivery sequence. Customer requirements can be noted and action can be taken on appointments, COD's, or other special handling requests. Customers with freight arriving on different inbound units can have their shipments consolidated, avoiding multiple deliveries or the problem of freight being misloaded in the delivery truck - requiring still more driver handling. Routes can be properly balanced and sequenced to reduce backtracking and avoid missed deliveries and pickups. The gains in time, fuel efficiency and customer satisfaction are enormous. They are only the beginning of the benefits you will receive with ROUSTRONIC 2000.</p> <p>ROUSTRONIC 2000 has a geo-coded database, so it knows every street and every address. A unique geo-coded grid system pinpoints the location of every customer whether they have used the service before or are first-time callers. The information can be displayed textually or, with an optional upgrade, via an onscreen map.</p> <p>ROUSTRONIC 2000 also provides continuous driver follow-up. As the dispatcher receives open pick-ups with recommended route and alternate route, he makes his choice and they are automatically sent to the correct driver. Confirmed pick-ups are recorded, "unassigned" and "unconfirmed" pick-ups are flagged in time for the dispatcher to take action. By giving the dispatcher - or anyone else with a screen - an accurate, up-to-the-minute view of all jobs, ROUSTRONIC 2000 makes it easy - whether it is real-time system management or just tracing orders.</p>			
<b>Platforms/ System Requirements:</b>	UNIX			
<b>Vendor:</b>	Carrier Logistics, Inc. 220 White Plains Road Tarrytown, NY 10591 U.S.A. Phone: (914) 332-0300 Fax: (914) 332-0949 e-mail: <a href="mailto:info@carrierlogistics.com">info@carrierlogistics.com</a>			
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 6/97	<b>Last update:</b> 4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.carrierlogistics.com">http://www.carrierlogistics.com</a>			

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe">University of Karlsruhe</a>				<a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe/">http://www.uni-karlsruhe.de/University_of_Karlsruhe/</a>
<b>STARS</b>				
<b>Category:</b>	Commercial	Vehicle Routing		
<b>Key Features:</b>	Smart Truck Assignment and Routing System: STARS determines optimal loading of products, routing, and schedules for delivery vehicles. As a daily operational tool, STARS can substantially reduce operating costs by minimizing driving distance and time while meeting stringent customer requirements.			
<b>Platforms/ System Requirements:</b>	Windows 95, Windows 98, Windows NT, Unix			
<b>Vendor:</b>	SAITECH, Inc. 1301 Highway 36 Hazlet, NJ 07733 U.S.A. Phone: (732) 264-4700 Fax: (732) 264-6015 e-mail: <a href="mailto:saitech@monmouth.com">saitech@monmouth.com</a>			
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 12/98	<b>Last update:</b> 2/1999
<b>Vendor's Homepage:</b>	<a href="http://www.saitech-inc.com">http://www.saitech-inc.com</a>			

<b>TESYS</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>					
<b>Platforms/ System Requirements:</b>					
<b>Vendor:</b>	Inform Software Corporation 2001 Midwest Road, Ste.100 Oakbrook, IL 60521 U.S.A. Phone: (800) 328-1489 Fax: (630) 268-7554 e-mail: <a href="mailto:george@iformusa.com">george@iformusa.com</a>				
<b>Price:</b>	call	<b>Source:</b>		<b>Last update:</b>	12/1997
<b>Vendor's Homepage:</b>					

## Further Information



[Previous](#) [Next](#)

Please mail any comment to [webmaster@wior.uni-karlsruhe.de](mailto:webmaster@wior.uni-karlsruhe.de)

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe/">http://www.uni-karlsruhe.de/University_of_Karlsruhe/</a>	
<b>Trapeze-Flex</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	FLEX is a scheduling and dispatching system that supports a more adaptable and responsive service by permitting fixed route vehicles to deviate from pre-defined patterns to accommodate passenger requests. The extent of trip deviations is controlled by slack times at bus stops and by restricting vehicles within service polygons that surround fixed routes.		
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT		
<b>Vendor:</b>	Trapeze Software Group Greenway-Hayden Loop Building A, Suite 200 Scottsdale, AZ 85260 U.S.A. Phone: (602) 627-8400 Fax: (602) 627-8411 e-mail: <a href="mailto:info@mail.trapezesoftware.com">info@mail.trapezesoftware.com</a>		
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage
<b>Vendor's Homepage:</b>	<a href="http://www.trapezesoftware.com">www.trapezesoftware.com</a>		
		<b>Last update:</b>	4/1999

## Further Information

### Trapeze-Flex's functionality:

#### Route Definition:

- Define patterns or route variations at time point, bus stop or street level
- Define and geocode time points and bus stops
- Plot new or altered transit routes

#### Client Registration:

- Track passengers throughout certification stages
- Scan original documents and photos for detailed client information
- Apply eligibility programs
- Store multiple client addresses for faster trip booking

#### Trip Reservations:

- Book subscription or demand trips with full access to passenger and trip data as well as passenger history
- Store multiple comment fields for additional reservation information including garage limitations

#### Scheduling:

- Schedule trips in real-time, batch, match assisted or background modes
- View multiple scheduling itineraries and runs interactively map
- Drag and drop manual scheduling for extra trips

#### Dispatching:

- Manage service on the actual day of service
- Identify and adjust service by responding to situations as they occur
- Minimize verbal communication between staff
- Record incidents detailing changes that happen to street itineraries
- Examine a comprehensive audit trail of activities relating to an entire trip

#### Complaints:

- Track complaints, commendations and suggestions
- Define user codes, access client and trip data, and document department or disciplinary actions

**Mobile Data Terminals (MDT):**

- Connect dispatchers and scheduler to drivers for live communication
- Send changes to drivers as their routes progress such as new trips and cancellations
- Update route progress enabling dispatchers to take corrective action where needed (eg vehicles running late)

**Automatic Vehicle Location (AVL):**

- View vehicles on the map in real time using global positioning satellite (GPS) technology
- Track schedule performance of vehicles
- Re-assign vehicles based on vehicle locations and schedule adherence

**Technology Standards**

The TRAPEZE® system is a true 32-bit software application built for Windows 95/NT. Features include:

- Three-tier client/server architecture
  - One common graphical interface
  - Embedded geographical information systems (GIS) functionality
  - TCPIP, IPX/SPX and named pipes protocols
  - ODBC and SQL compliant database management systems
  - Configurable for local and wide area networks
  - Enabled for the Internet
  - Configurable for single-user and multi-user environments
  - Standard and ad hoc reporting
  - Enabled for other technologies (automatic vehicle location and computerized voice)
-

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/Uni/">University of Karlsruhe</a>		<a href="http://www.uni-karlsruhe.de/Uni/">http://www.uni-karlsruhe.de/Uni/</a>	
<b>Trapeze-Pass</b>			
<b>Category:</b>	Commercial	Vehicle Routing	
<b>Key Features:</b>	<p>Trapeze-Pass is an innovative demand-responsive scheduling and dispatching software system. This advanced software enables you to effectively and efficiently accomplish your routine tasks, including:</p> <ul style="list-style-type: none"> <li>• registering passengers</li> <li>• creating bookings</li> <li>• scheduling passengers to vehicles</li> <li>• dispatching vehicles and drivers</li> <li>• recording trip events</li> <li>• geocoding locations</li> </ul> <p>You can choose to manually schedule passengers and dispatch vehicles, or have the system automatically do the work for you.</p>		
<b>Platforms/System Requirements:</b>	Windows 95, Windows NT		
<b>Vendor:</b>	<p>Trapeze Software Group                  Greenway-Hayden Loop                  Building A, Suite 200                  Scottsdale, AZ 85260                  U.S.A.                  Phone: (602) 627-8400                  Fax: (602) 627-8411                  e-mail: <a href="mailto:info@mail.trapezesoftware.com">info@mail.trapezesoftware.com</a></p>		
<b>Price:</b>	call	<b>Source:</b>	vendor's homepage
		<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.trapezesoftware.com">www.trapezesoftware.com</a>		

**Further Information**

**Trapeze-Pass's functionality:**

**System Map:**

- Colorful mapping tool enables easy geocoding – the longitude and latitude coordinates are automatically calculated
- Customize the map to show specific map colors, route colors and street names

**Client Registration:**

- Apply eligibility programs such as American Disabilities Act (ADA) to paratransit service and track passengers' certification status
- Save scanned documentation and photos for detailed client information

**Trip Booking:**

- Book subscription or demand trips with full access to passenger and trip data as well as passenger history
- Multiple comment fields for storing additional reservation information
- Passenger suspension notification

**Scheduling:**

- Automatic assignment of recurring subscription trips (batch mode)
- Automatically match subscription clients on the same recurring trips (match mode)
- View multiple scheduling itineraries and runs interactively on the map

- Drag and drop manual scheduling

**Dispatching:**

- Monitor vehicles, runs and trips
- Identify and adjust service by responding to situations as they occur
- Record incidents detailing any changes that happen to street itineraries

**Enhancement Series:**

- Augment your capabilities with one or all of our enhancement products.

**Complaints:**

- Track complaints, commendations and suggestions
- Define user codes, access client and trip data, and document department or disciplinary actions

**Certification:**

- Track multiple funding sources
- Implement a complete certification process

**Mobile Data Terminals (MDT):**

- Connect dispatchers and scheduler to drivers for live communication
- Send changes to drivers as their routes progress such as new trips and cancellations
- Update route progress enabling dispatchers to take corrective action where needed (eg vehicles running late)

**Automatic Vehicle Location (AVL):**

- View vehicles on the map in real time using global positioning satellite (GPS) technology
- Track schedule performance of vehicles
- Re-assign vehicles based on vehicle locations and schedule adherence

**Computerized Voice:**

- Support trip confirmations, cancellations, bookings and passenger call outs without human intervention at the reservation center

**The Malteze Database:**

Malteze database is the foundation architecture for all modules in the TRAPEZE® system. Developed by Trapeze Software Group, Malteze enables the TRAPEZE® applications and components to seamlessly share data.

**Technology Standards**

The TRAPEZE® system is a true 32-bit software application built for Windows 95/NT. Features include:

- Three-tier client/server architecture
  - One common graphical interface
  - Embedded geographical information systems (GIS) functionality
  - TCPIP, IPX/SPX and named pipes protocols
  - ODBC and SQL compliant database management systems
  - Configurable for local and wide area networks
  - Enabled for the Internet
  - Configurable for single-user and multi-user environments
  - Standard and ad hoc reporting
  - Enabled for other technologies (automatic vehicle location and computerized voice)
-

Software Library for Operations Research - <a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe">University of Karlsruhe</a>					<a href="http://www.uni-karlsruhe.de/University_of_Karlsruhe/">http://www.uni-karlsruhe.de/University_of_Karlsruhe/</a>
<b>Truckstops for Windows</b>					
<b>Category:</b>	Commercial		Vehicle Routing		
<b>Key Features:</b>	TruckStops handles nearly all routing situations quickly and accurately and gives you excellent results. It takes into account a wide range of data about your stops and vehicles, as well as all of the associated expenses in order to model your operation accurately. While adhering to the rules you set, TruckStops gives you the lowest-cost, best-service routes. Given a realistic picture of your operation, TruckStops then knows how to improve it. You can tailor TruckStops to your individual operation by using over 90 different Options settings and hundreds of data items. You can set the exact amount of time you want it to spend improving the routes as well as the number of time windows (up to 10) you would like to have for each customer. With TruckStops, there is no need for expensive customization or extensive implementation, and initial setup is relatively quick and smooth, often averaging four to eight weeks.				
<b>Platforms/System Requirements:</b>	Windows 95, Windows 98, Windows NT Pentium processor with 16 MB of RAM, and 50 MB of hard disk space or better				
<b>Vendor:</b>	Micro Analytics, Inc. 2200 Clarendon Blvd., Suite 1002 Arlington, VA 22201 U.S.A. Phone: (703) 841-0414 Fax: (703) 527-1693 e-mail: <a href="mailto:info@bestroutes.com">info@bestroutes.com</a>				
<b>Price:</b>	call	<b>Source:</b>	OR/MS Today 6/97	<b>Last update:</b>	4/1999
<b>Vendor's Homepage:</b>	<a href="http://www.bestroutes.com">http://www.bestroutes.com</a>				

## Further Information

Click [here](#) to obtain information on how to get a demo version.

TruckStops requires three basic types of data: Information On Your Stops — A Stop file includes all the necessary information on your customers and their requirements, such as: stop name, address, identification numbers and latitude and longitude coordinates when you have to be there (time windows) how long it will take to service that customer (fixed stop time, unload rates) what you are delivering or picking up (quantity) You can identify special customer requirements such as headloads or tailloads, or match a certain vehicle with a certain stop. You can also enter important instructions or directions for the drivers which can be produced on route or truck reports produced through TruckStops' Report Writer. Information On Your Vehicles — TruckStops allows you to define each vehicle's individual characteristics independently. This means you can have as many vehicle types as you have physical vehicles. The Truck file contains items about the vehicles such as: cost factors, including cost per mile, cost per hour and overtime work rules origin and destination maximum capacity of the vehicle in user-defined units of measure depending on your operation Also in the Truck file, you can set vehicles to be redispached, schedule work hours and identify special equipment such as a lift gate the different vehicles. General Information (Parameters) — The Parameter file includes defaults and data which is not specific to individual stops or vehicles, for example, if you only have one warehouse for all your vehicles, that location would be in the parameter file rather than in each individual vehicle's file. Other Data Available Additional data you can load into TruckStops includes: Maps — A variety of road maps can be used in TruckStops, from detailed street maps for local delivery to more general maps for over-the-road routing. U.S. and state highway maps are available directly from MicroAnalytics. Barriers — You can enter data about geographic barriers (rivers, lakes, etc) that may affect travel between stops. You tell TruckStops where the barrier is located and the possible transit points. Road Networks — To further enhance the accuracy of travel times and

mileage, you can use ALK Associates' PC\*MILER® and PC\*MILER®/Streets through a seamless interfaced developed to work with TruckStops 2.0. The system will use the travel distances and time (all or selected ones) from PC\*MILER instead of the distances and times it calculates. Data Transfer Besides our interface with ALK Associates' PC\*MILER® and PC\*MILER ®/Streets, you can export or import any data from TruckStops via a fixed ASCII format. This enables users to quickly and easily download orders from their AS400 or any order-entry system into TruckStops, run the routes, and then export the new routes either back in the operation's main computer system, or into onboard computer.

---

## ***Control De Reuniones***

**Fecha: 23/3/99**

**Participantes: Ing. Omar Viera (OV), Ing. José Comas (JC), A/S Carlos Pedezert (CP), A/S Javier Barreiro (JB), A/S Nicolás Sosa (NS)**

### **Temas tratados:**

- Objetivo y concepto general del proyecto.
- Generalidades sobre la forma de trabajo, reuniones a realizarse, comunicación entre los estudiantes y responsables del proyecto.
- OV como responsable académico del proyecto será el encargado de controlar el avance del mismo y de analizar con los estudiantes los algoritmos a utilizar para resolver los diferentes problemas.
- JC como responsable por la empresa solicitante del proyecto será el encargado de determinar los requerimientos del mismo y las herramientas a utilizar.
- El resultado final deberá ser un paquete de software que cumpla los requerimientos del proyecto, un informe del mismo y del desarrollo del proyecto y una demostración del producto.
- Próximas actividades a realizar: Buscar información en internet sobre proyectos similares ya existentes, estudiar bibliografía sobre ruteo de vehículos, elaborar plan de trabajo.

**Fecha: 22/4/99**

**Participantes: OV, JC, CP, JB, NS**

### **Temas tratados:**

- Presentación del Plan de Trabajo inicial del proyecto. Deberá ser ampliado para la próxima reunión.
- Productos ya existentes a considerar (ArcLogistic, AvRoute, AgeMap, RouteSmart, RoadShow).
- Definición de la lista de problemas a tratar: TSP, VRP, MDVRP (todos con y sin las variantes de flota heterogénea y ventanas de tiempo), CPP, DARP.
- Determinación de herramientas de desarrollo a utilizar: MapObjects desde Visual Basic para la internase gráfica con el usuario. Los datos se recibirán en archivos .shp (datos geográficos), .dbf y ASCII (información sobre los vehículos). El motor de la base de datos y la resolución de los algoritmos se realizará en C++.
- Próximas actividades a realizar: Estudio y elección de algoritmos para TSP y VRP. Estudio de técnicas para resolver los algoritmos de MDVRP (asignar y luego rutear, asignar y rutear simultáneamente, etc.). Informe preliminar que incluya: justificación de la lista de problemas a tratar, descripción detallada de cada problema, elección y justificación de algoritmos para los problemas TSP y VRP, elección de los lenguajes de programación, arquitectura general del sistema.

**Fecha: 19/5/99**

**Participantes: OV, CP, JB, NS**

### **Temas Tratados:**

- Presentación del Plan de Trabajo actualizado del proyecto.
- Presentación y revisión primaria del informe solicitado en la reunión anterior. Falta descripción del problema DARP y se debe ampliar la introducción.
- Avance del proyecto.
- Solicitar apoyo a JC sobre problemas surgidos con el MapObjects, estructura de datos, requerimientos.
- Técnicas para los problemas de MDVRP. La técnica de asignar y rutear al mismo tiempo para problemas de gran tamaño puede llegar a ser muy lento. Seguir investigando.
- Próximas actividades a realizar: Ampliar informe. En la medida que sea posible comenzar con la implementación de los algoritmos. Aprendizaje de herramientas y desarrollo de interfase gráfica.

**Fecha: 29/6/99**

**Participantes: OV, JC, CP, NS**

**Temas Tratados:**

- Avance del proyecto.
- Problemas surgidos con la herramienta MapObjects al no tener la versión correcta y por lo tanto con el desarrollo de la interfase gráfica. Se resuelve seguir tratando el tema pero dejarlo para más adelante si frena el avance natural del proyecto. JC entregará nueva versión del MapObjects.
- Técnicas a utilizar para los problemas de MDVRP. Se resuelve utilizar asignar primero y luego rutear.
- Profundización en el tema de la conexión de la interfase gráfica con los algoritmos. En principio se realizará a través de archivos txt. La interfase gráfica debe permitir levantar los datos ingresados mediante archivos .shp ,dbf, txt y modificar algunos de dichos datos.
- Estructura del informe final:
  - Abstract (25 líneas aprox.)
  - Índice
  - Introducción
  - CAP 1 VRP
  - Etc.
  - CAP m CPP
  - CAP m+1 DARP
  - CAP m+2 Implementación y Testeo
  - CAP m+3 Resultados
  - CAP m+4 Trabajos futuros
  - Bibliografía
  - Apéndices (Información obtenida de internet, plan de trabajo y avances, control de reuniones)
- Material a entregar al finalizar el proyecto: Informe final, manual de usuario, CD con el producto.
- Próximas actividades a realizar: Continuar con el desarrollo de la interfase gráfica, ampliar y adaptar el informe existente a la nueva estructura definida, seguir estudiando conexión de la interfase gráfica con los algoritmos, comenzar el desarrollo de los algoritmos.

**Fecha: 20/8/99**

**Participantes: OV, JC, CP, JB, NS**

**Temas tratados:**

- Avance del proyecto. Se resuelve actualizar el plan del proyecto al estado de situación actual.
- En la realización de los algoritmos se priorizará su reusabilidad y rapidez frente a la utilización de memoria. La idea es hacer un paquete genérico que luego pueda ser mejorado y adaptado a las diferentes necesidades de los posibles clientes por lo que es primordial con los algoritmos sean reusables y no sean lentos por motivos de su diseño básico. El tema de la utilización de memoria si bien debe ser tenido en cuenta no es considerado primordial.
- Visualización de un ejemplo de interfase gráfica similar a la requerida para el proyecto. Se establece que el usuario ingresara los archivos .shp a ser utilizados para el programa pero los mismos no podrán ser modificados durante la ejecución del programa. Se solicitaron archivos .shp de ejemplo para poder realizar las pruebas respectivas.
- Necesidad de argumentar en el informe final todas las decisiones tomadas. Porqué se eligió la lista de problemas utilizados, algoritmos utilizados, etc.
- Próximas actividades a realizar: Preparar para el 1º de setiembre un “diseño” de los algoritmos a utilizar. Con esto nos referimos a que para cada caso, para cada problema determinar que algoritmo será utilizado, su pseudo-código y estructura de datos a utilizar. La idea es dejar todo listo para el 1º de setiembre iniciar de lleno la implementación. Próxima reunión martes 31/8 con informe y plan del proyecto actualizado.

**Fecha: 2/9/99**

**Participantes: OV, JC, CP, JB, NS**

**Temas tratados:**

- Dudas sobre correcciones del informe.
- Dudas sobre la definición de los problemas y los algoritmos a utilizar para su resolución.  
Se establece que en el VRP clásico con flota homogénea la cantidad de vehículos no es sabida en un principio y no es considerada al hallar la solución. La solución es la que determina la cantidad de vehículos necesarios. En el problema con flota heterogénea se debe saber con cuántos vehículos de cada tipo se cuenta para brindar la solución adecuada a dicha restricción.  
En el CPP se determina que en cada arco pueden haber varios clientes con demanda y que la información brindada es la demanda de cada cliente y no la demanda total del arco. Con esto el problema queda planteado como satisfacer la demanda de todos los clientes los cuales están ubicados en los arcos sin necesidad de recorrer completamente todos los arcos.
- Discusión y presentación de las estructuras de datos a utilizar y pseudo-código de los algoritmos.

**Fecha: 27/9/99**

**Participantes: OV, NS**

**Temas Tratados:**

- Problema DARP. Caso a resolver y algoritmo a utilizar.

**Fecha: 18/11/99**

**Participantes: OV, CP, JB, NS**

**Temas Tratados:**

- Avance del proyecto.
- Dudas en implementación.
- Estimación y planificación del resto del proyecto. Se pretende finalizar la implementación para mediados de diciembre y presentar una primera versión del informe final los primeros días de enero.

**Fecha: 9/2/00**

**Participantes: OV, JC, CP, JB, NS**

**Temas tratados:**

- Avance del proyecto. Se posterga la presentación del mismo para Marzo del 2000.
- Problemas con el software en la conexión entre la interfase gráfica y los algoritmos. El problema se presenta con la utilización de las dll generadas. Posibles soluciones y alternativas.
- Revisión del informe.
- Material a entregar al finalizar el proyecto:
  - Informe final.
  - Manual de usuario.
  - 1 CD para I.C.A. S.A. con toda la información del proyecto, es decir documentos, archivos fuentes y ejecutables con el software desarrollado.
  - 1 CD para la facultad con toda la información del proyecto menos los archivos fuente.
- Próximas actividades a realizar: Preparar para el 14 de febrero nueva versión del informe final con las correcciones vistas en la reunión.

**Fecha: 28/2/00**

**Participantes: OV, NS**

**Temas Tratados:**

- Ver correcciones al informe y pendientes.

**Fecha: 10/3/00**

**Participantes: OV, NS**

**Temas Tratados:**

- Ver correcciones al informe y pendientes.

**Fecha: 15/3/00**

**Participantes: OV, NS**

**Temas Tratados:**

- Ver correcciones al informe y pendientes.

## ***Plan De Trabajo Y Avance Del Proyecto***

### **Plan de Trabajo Inicial del Proyecto.**

Fecha: 19/4/99

Tareas:

- |                                    |                          |
|------------------------------------|--------------------------|
| ➤ Revisión bibliográfica.          | Del 22/3/99 al 7/5/99    |
| ➤ Definición de requerimientos.    | Del 22/4/99 al 17/5/99   |
| ➤ Definición de funciones.         | Del 10/5/99 al 28/5/99   |
| ➤ Diseño de las funciones.         | Del 24/5/99 al 11/6/99   |
| ➤ Aprendizaje de MapObjects.       | Del 07/6/99 al 16/6/99   |
| ➤ Desarrollo de interfase gráfica. | Del 14/6/99 al 09/7/99   |
| ➤ Implementación.                  | Del 05/7/99 al 15/10/99  |
| ➤ Testeo.                          | Del 18/10/99 al 05/11/99 |
| ➤ Preparación de informe final.    | Del 08/11/99 al 26/11/99 |

### **Avance del proyecto**

Fecha: 19/4/99

Se está cumpliendo con el cronograma establecido.

### **Plan de Trabajo del Proyecto**

Fecha: 17/5/99

Tareas:

- |  |                          |
|--|--------------------------|
| ➤ Revisión bibliográfica.  | Del 22/3/99 al 07/5/99   |
| ➤ Definición de requerimientos.  | Del 22/4/99 al 07/5/99   |
| ➤ Def. y diseño de funciones,<br>arquitectura del sistema y base de datos                | Del 10/5/99 al 28/5/99   |
| ➤ Aprendizaje de herramientas para interfase gráfica.                                    | Del 24/5/99 al 11/6/99   |
| ➤ Desarrollo de interfase gráfica.   | Del 14/6/99 al 09/7/99   |
| ➤ Aprendizaje de herramientas para desarrollo y<br>su conexión con la interfase gráfica. | Del 05/7/99 al 13/7/99   |
| ➤ Implementación.  | Del 14/7/99 al 17/9/99   |
| • TSP  | 14/7/99 al 27/7/99       |
| • VRP  | 26/7/99 al 06/8/99       |
| • MDVRP  | 09/8/99 al 20/8/99       |
| • MDVRPTW  | 23/8/99 al 03/9/99       |
| • Dist. Period.  | 30/8/99 al 10/9/99       |
| • CPP y DARP   | 06/9/99 al 17/9/99       |
| ➤ Testeo.  | Del 02/8/99 al 08/10/99  |
| ➤ Preparación de informe final.  | Del 11/10/99 al 26/11/99 |
| ➤ Preparación de presentación final.   | Del 28/10/99 al 15/12/99 |

La división realizada dentro de la tarea Implementación pretende ser una guía de referencia y no un plan de trabajo detallado de cómo deberá ser desarrollada la implementación.

### **Avance del proyecto**

Fecha: 19/5/99

Se nota un pequeño retraso en el avance del proyecto. Dicho retraso es considerado leve.

Aún no se ha completado la etapa de definición de requerimientos.

### **Avance del proyecto**

Fecha: 19/6/99

Gran retraso en el avance del proyecto. La etapa de aprendizaje de herramientas para el desarrollo de la interfase gráfica debería estar finalizada y recién está comenzando. Este retraso se debe en parte a no haber conseguido la versión correcta del programa (MapObjects) a utilizar y haber realizado pruebas con uno incorrecto.

### Avance del proyecto

Fecha: 29/7/99

Se sigue notando un importante retraso en el avance del proyecto.

Debería estar finalizada la interfase gráfica y estar en pleno desarrollo de los algoritmos pero la realidad indica que se está comenzando con la implementación de los algoritmos y la interfase gráfica no se ha finalizado.

La secuencia de etapas del proyecto prevista en el plan de trabajo del proyecto se ha modificado sustancialmente ya que se han repartido distintas tareas entre los integrantes del grupo. Dichas tareas en un principio eran consideradas etapas diferenciadas en el tiempo pero ahora son realizadas simultáneamente. Por ejemplo estaba previsto comenzar con la implementación de los algoritmos una vez finalizada la interfase gráfica pero dado el atraso que se produjo en la implementación de la interfase gráfica se resolvió realizar ambas tareas en paralelo.

### Actualización del Plan de Trabajo del Proyecto

Fecha: 21/8/99

Tareas:

- |   |                          |
|---|--------------------------|
| ➤ Desarrollo de interfase gráfica.                              | Hasta 10/10/99           |
| ➤ Implementación.   | Del 01/09/99 al 10/10/99 |
| ➤ Preparación de informe final.                                 | Del 11/10/99 al 07/11/99 |
| ➤ Preparación de presentación final y cambios al informe final. | Del 08/11/99 al 17/12/99 |

Implementación incluye el desarrollo de todos los módulos y su testeo.

### Actualización del Plan de Trabajo del Proyecto

Fecha: 21/11/99

- 18/11 – 24/11

Terminar y dejar como módulo independiente VRP, MDVRP, VRPTW, MDVRPTW y captura de datos enviados desde la interfase gráfica.

Comenzar con VRP flota heterogénea y DARP.

Terminar primera etapa de la interfase gráfica, la cual incluye: ingreso de datos por parte del usuario para los problemas VRP, MDVRP, VRPTW y MDVRPTW; generación de archivos txt para comunicación con los algoritmos; visualización de resultados de los algoritmos.

- 25/11 – 10/12

Terminar VRP flota heterogénea y DARP.

Realizar CPP.

Interfase gráfica para dichos problemas.

Comenzar testeos generales y comunicación de la interfase gráfica con los algoritmos.

- 11/12 – 18/12

Finalizar CPP y posibles pendientes de otros algoritmos.

Testear todos los módulos.

Integración total del sistema.

- 19/12 – 31/12

Realizar primera versión del informe final.

Testeo general del sistema.