DESIGN AND DEVELOPMENT OF A VEHICLE ROUTING SYSTEM UNDER CAPACITY, TIME-WINDOWS AND RUSH-ORDER RELOADING CONSIDERATIONS

A Thesis

by

GOPALAKRISHNAN EASWARAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2003

Major Subject: Industrial Engineering

DESIGN AND DEVELOPMENT OF A VEHICLE ROUTING SYSTEM UNDER CAPACITY, TIME-WINDOWS AND RUSH-ORDER RELOADING CONSIDERATIONS

A Thesis

by

GOPALAKRISHNAN EASWARAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Sila Çetinkaya
(Chair of Committee)

Guy L. Curry
(Member)

F. Barry Lawrence
(Member)

Brett A. Peters
(Head of Department)

August 2003

Major Subject: Industrial Engineering

ABSTRACT

Design and Development of a Vehicle Routing System under Capacity, Time-Windows and Rush-Order Reloading Considerations. (August 2003)

Gopalakrishnan Easwaran, B.E., PSG College of Technology, India

Chair of Advisory Committee: Dr. Sila Çetinkaya

The purpose of this research is to present the design and development of a routing system, custom developed for a fence manufacturing company in the continental US. The objective of the routing module of the system is to generate least cost routes from the home-center of the company to a set of delivery locations. Routes are evolved for a set of customer locations based on sales order information and are frequently modified to include rush orders. These routes are such that each delivery is made within a given time window. Further, total truckload of all delivery locations over any particular route is not allowed to exceed the weight and volume capacities of the truck.

The basic system modules such as user interface functions and database are designed using MS Access 2000. An interface module to retrieve data from existing ERP system of the company is developed to import pick-ticket information. A customer inter-distance maintenance module is designed with the abilities of a learning tool to reduce information retrieval time between the routing system and the GIS server. Graphical User Interface with various screen forms and printable reports is developed along with the routing module to achieve complete system functionality and to provide an efficient logistics solution.

This problem, formulated as a mixed-integer program, is of particular interest due to its generality to model problem scenarios in the production shop such as job-shop scheduling, material handling. This problem is coded and solved for instances with different input parameters using AMPL/CPLEX. Results of test runs for the company data show that the solution time increases exponentially with the number of customers. Hence, a heuristic approach is developed. Sample runs with small instances are solved for optimality using AMPL/CPLEX and are used to compare the performance of the heuristics. However, test runs solved using the heuristics for larger instances are compared with the manual solution. The comparison shows a considerable cost savings for heuristic solutions. A what-if analysis module is implemented to aid the dispatcher in choosing input parameters based on sensitivity analysis. In conclusion, further improvement of the routing system and future research directions are proposed.

ACKNOWLEDGMENTS

I sincerely thank my advisor Dr. Sila Çetinkaya for her kind guidance, encouragement and advice during this research and all through my studies here at Texas A&M University. My humble and sincere thanks to Dr. F. Barry Lawrence for providing me with this excellent opportunity to pursue real time research for a company. I sincerely acknowledge and appreciate his effort in imparting to me his practical skills in project management and project implementation. I also thank Dr. Guy L. Curry for his guidance during my research and for serving on my committee. Special thanks to Mr. Ramesh Krishnamurthy, Mr. Bharani Nagarathnam, and Mr. Abhishek Kanungo for their support during this research. Thanks also to Mr. Tim Miller, Vice President of Operations of the company for his constructive criticism and great comments during the presentations and conference calls. Finally, I owe a debt of gratitude to the managers, engineers, and technicians at the regional distribution center of the company, for sparing their expensive time to provide valuable information and support during this research and making us feel at home during field stays at their place.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE Page

INTRODUCTION

The purpose of this research is to design and develop a Routing System, customized for a fence manufacturing company in the continental US. In general, vehicle routing and scheduling forms a primary operational issue in distribution management. Developing effective routes for capacitated vehicles through a set of customer locations with delivery schedules, results in benefits of cost savings, customer satisfaction and efficient resource management.

Distribution logistics forms the second largest business process next only to manufacturing and inventory control, for the fence manufacturing company. The company is involved in manufacturing and distribution of fence products and it owns about 58 regional distribution centers operating with their own fleet of different sizes and capacities. The regional distribution center (RDC) considered for the research is located at Dallas, TX. The company ships about thirty five customer orders everyday around its geographic location. The RDC owns and operates identical commercial trucks (18-wheelers). The company has an integrated distribution enterprise resource planning (ERP) system for its planning and operations. On a typical day, the company accumulates sales orders through phone calls and fax messages till 3:00pm for the next day's deliveries. The dispatcher then clusters the corresponding pick-tickets manually to form delivery routes based on time-windows and vehicle capacities. Once the clustering is done, the routes are released to the staging area and instructions are given for loading the customer orders into corresponding fleet.

However, the company still processes sales orders received between 3:00pm to 5:00pm as rush orders. The RDC's past data shows that the company receives about five to ten rush orders per day. By the end of the day, the dispatcher reroutes the orders by appending or

---

This thesis follows the style and format of Transportation Research Part B.

inserting or adding a new route, to accommodate the rush orders. This causes a heavy unloading and reloading of trucks, resulting in overtime costs and energy. The company wishes to reduce this nervousness due to rush order arrivals and is interested in finding a trade off between the reloading costs and the cost the company would incur if the rush orders were just appended or routed in a separate truck.

The above mentioned problem can be viewed and modeled as an extension to the classical capacitated vehicle routing problem with time-windows (CVRPTW). However, research focus is required in modeling the problem for loading nervousness due to rush-orders and finding an effective solution procedure to solve the problem to optimality. Further, the problem understudy is interesting for its applicability. Consider a job-shop with ready time and due dates for jobs with sequence dependent setup times. Consider an objective to minimize the makespan of the job-shop. Even with a single machine case, this problem is classified under the category of strongly NP-Hard problems by Pinedo, 2002. For a detailed study on the sequence dependent setup times, see Bianco, et al., 1988; Tang, 1990; Wittrock, 1990. The setup time between two jobs is analogous to the distance between any two delivery locations. The ready time and due dates are analogous to the open time and close time (time-window) of the delivery locations. The machine capacity (processing time) is equivalent to the vehicle capacity. Thus, the sequence dependent setup time problem is analogous to the vehicle routing problem (VRP) with time-windows and capacity restrictions. Similarly, material handling problem that requires routing of handling devices to transport work in process (WIP) between machines according to a predetermined schedule can be viewed as a direct analogy to the CVRPTW within a manufacturing facility. Thus, the mathematical problem understudy can be generalized and used to model and solve scenarios in manufacturing, material handling and distribution of customer orders.

This thesis is divided into seven major sections. The following section describes the issues involved in design of the system, the architecture and the information flow. The third section details the implementation issues such as design of database, the graphical user interface, and the geographic information system interface. The fourth section outlines a review on the modeling approaches and solution techniques for the vehicle routing problem. The fifth section deals with the algorithm design to solve the CVRPTW with rush order processing (CVRPTWRO). Two heuristic methods are outlined in this section. The sixth section details the computational results of the heuristics. It further presents a comparison between the heuristics and optimal solution for small instances of the problem. However, larger instances of the problem solved using the heuristics are benchmarked against the manual solution (since optimal solutions cannot be determined with in a reasonable time period). The company data such as order information and customer inter-distances, the routing schedule obtained from the routing system, and the AMPL/CPLEX branch and bound search results are presented in the appendix.

ROUTING SYSTEM DESIGN

The present work is a customization for the requirements of the fence manufacturing company, having one of its regional distribution centers at Dallas, TX. The system is designed to aid the dispatcher in routing the orders by providing cost effective routes at reduced time and effort, through process automation and integration. The following section describes the functions and performance of the routing system and the constraints that governed its development. This section further highlights the subsystem modules, their interrelationships and input-output data. The system specification also describes the information (data and control) that is input into and output from the system.

System Objectives

In short, the objective of the work can be stated as “to design and develop a system for vehicle routing”. The system is designed to achieve the following features.

- Development of a *data-extraction interface* for the company’s existing ERP system, which will provide order and customer information to the downstream applications of the routing system.
- Development of *databases* to have customer delivery locations with street level physical addresses, the GIS input parameter string of the delivery locations sorted by customer numbers, individual part weight, part classification, and other specifications such as part dimensions and bundle size.
- Development of a special purpose *vehicle routing algorithm* to generate efficient routes for the selected set of customer delivery locations from the home center with the

objective of minimizing the cost under capacity, time-windows and rush-order reloading considerations.

- Development of a *Re-optimization* module to enable the dispatcher to interact with and modify the generated routes to have additional constraints.

Constraints Governing the System Design

Since the software tool is custom built for the requirements of the company, it has the following constraints.

- The software is a PC based tool that requires Microsoft Windows 98 (or above) for the operating system.
- The customer inter-distances obtained through the GIS, PCMILER STREETS 3.0 (see reference), will be used as data for the routing algorithm. The interactions with the PCMILER Server will be performed through a module built in VC++.
- The software tool is developed using Microsoft Access 2000, with Visual Basic Application (VBA), and hence the system requires the package for its regular operation.
- The Open Data Base Connectivity (ODBC) Data Source Name (DSN) setup for the Microsoft Access to access the company's database is required. This provides access security through login and password setup procedure.
- The design of the database and the system is based on the following information provided by the company either as parameters, or as data.
    - Pick-Slip Information containing the details of the customer order such as part numbers, quantity, and required time of delivery.
    - Part information such as part weight and quantity
    - Average travel speed of trucks

  - Average unloading time
  - Maximum wait time to deliver an order due to time window constraints
  - Fixed and variable costs for each route.

System Architecture Context Diagram

The architecture context diagram (ACD) illustrates various modules of the routing system with its interaction with the departments of the organization. The context diagram is shown in Fig. 1. The architecture of the routing system can be viewed as comprising of the following five major processing segments.

Input Processing

The major input requirements for the routing system have been identified as coming from the sales order processing (the pick-slip information), resource management (truck/carrier specifications), warehouse management (part specifications), personnel management (federal rules governing the driving operations such as maximum time for driving), and the GIS - PCMILER STREETS 3.0 (customer inter-distances). Based on the nature of the data obtained from various departments, they are classified under three domains.

- *Data Domain* containing the (1) Customer delivery addresses (street level address), (2) Time window information, (3) Order information, (4) Truck details and requirement of a material handling device at the delivery locations.
- *Parameters domain* consisting of the parameters of the algorithm such as (1) average travel speed of the truck, (2) average unloading time, (3) maximum wait time before a customer delivery in order to satisfy the time window constraints, (4) mileage costs and other fixed/variable costs.

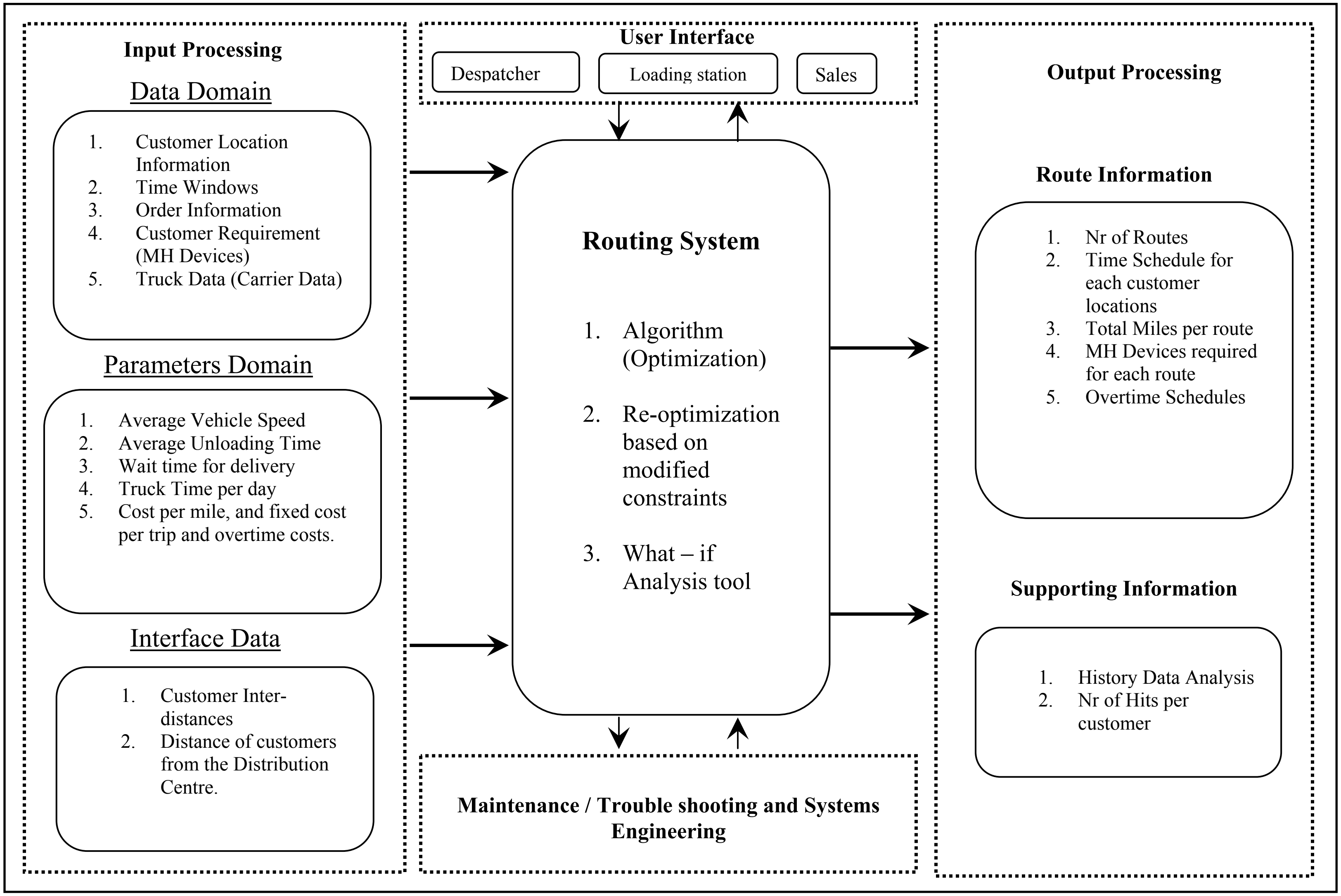


Fig.1. Input – output architecture diagram for the vehicle routing system

- *Interface domain* containing (1) the GIS Server to obtain the customer inter-distances and (2) distances of the customer delivery locations from the distribution center.

Output processing

In the output processing, two types are information are reported for further use. They include the *route information* having the number of routes, time schedule, and cost information and the *supporting information*, which is useful in deriving the customers who requests delivery for more than a critical number and which enables to store their inter-distances for efficient data management. The generated routes are provided as output to the staging station, the loading dock, and the personnel scheduling departments. Further, the dispatcher can modify the presented routes and do a what-if analysis to find the impact of modifications on the cost of routing.

User Interface

Even though it has been planned to automate majority of the processing and control activities of the routing system, provision has also been given for the user to interact with the system. Particularly, the customer delivery locations and time window constraints requires user interaction.

Maintenance

The routing system designed and developed needs to be maintained during its usage. This includes the periodic maintenance of the customer inter-distance database, adding additional features to the software or hardware, and updating the versions of the supporting software packages.

Information Flow Diagram

The information flow diagram in Fig. 2 illustrates the flow of data in the software and its interaction with different entities. First, the *order information* is queried through the dial-up network or through a data line connection from the company. The data is stored in the *transaction* table and is verified for their corresponding customer delivery locations. New customer, their detailed address and parts addition, their specifications are required as input to the *Data Domain*.

The Customer delivery locations are sent to the GIS Server to get their inter-distances. Then the data is stored in the *transaction database* for the Algorithm to generate an efficient route based on the parameters available in the *parameter domain.*

The routes generated are presented to the user in a GUI where the routes can be altered and the constraints of the optimization problem can be modified. Then, the Re-optimization module generates a new set of routes based on the new constraints. Further, a what-if analysis, by varying the parameters such as wait time or truck time, can also be performed. A report containing the generated routes, with detailed time schedules and delivery addresses, is generated for printouts and further processing to load the orders in the truck.

The details are stored in a history database, where from frequency of delivery requests from customers can be studied. Since the volume of information handled during the inter-distance generation process is very high and since the interface with the GIS to generate the distances takes much time, the *customer inter-distance learning / maintenance module* is designed to enable efficient data management. The design features and operation of each of the individual modules and databases are discussed in detail in the subsequent section.

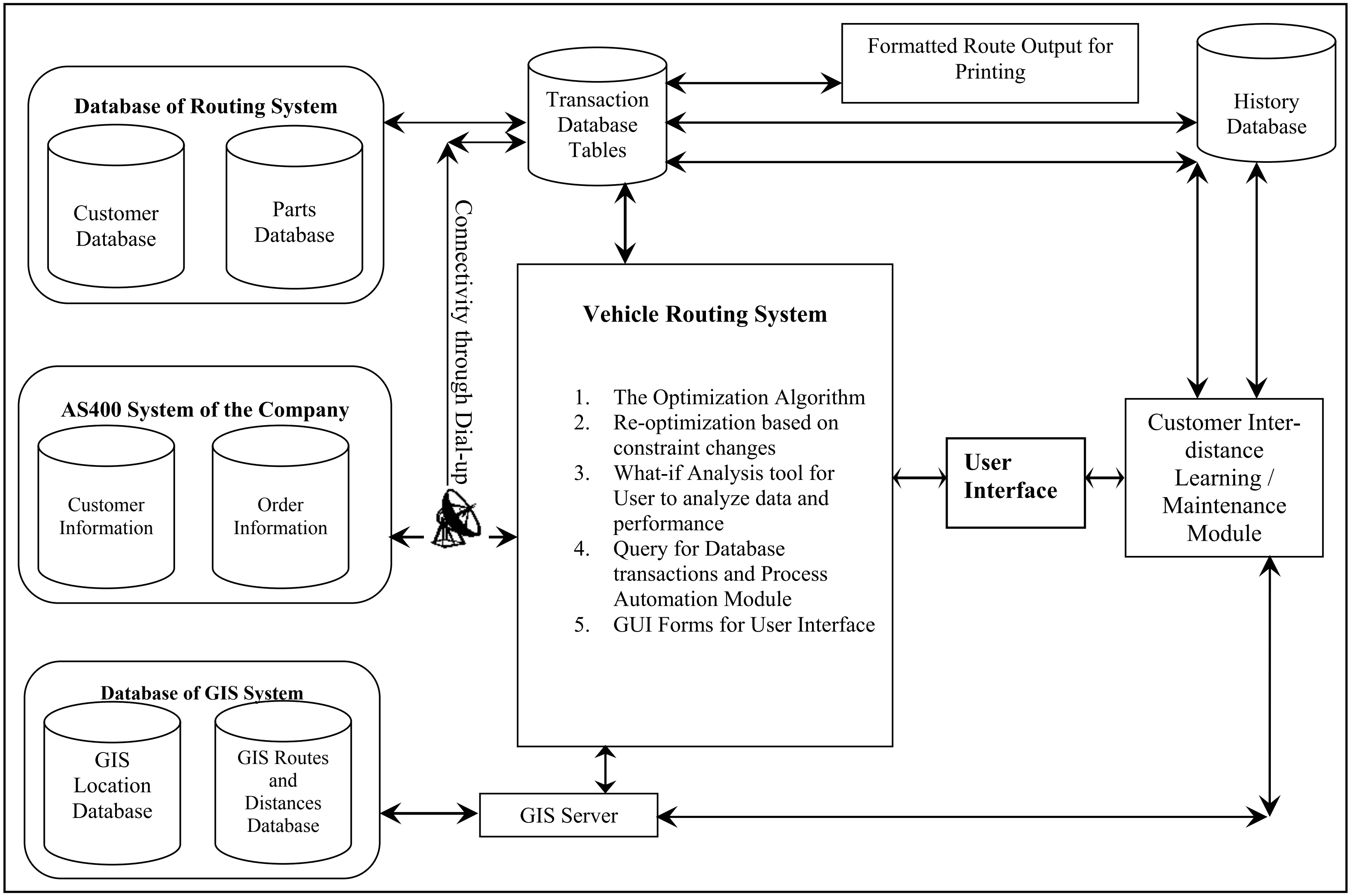


Fig. 2. Information flow diagram for the vehicle routing system

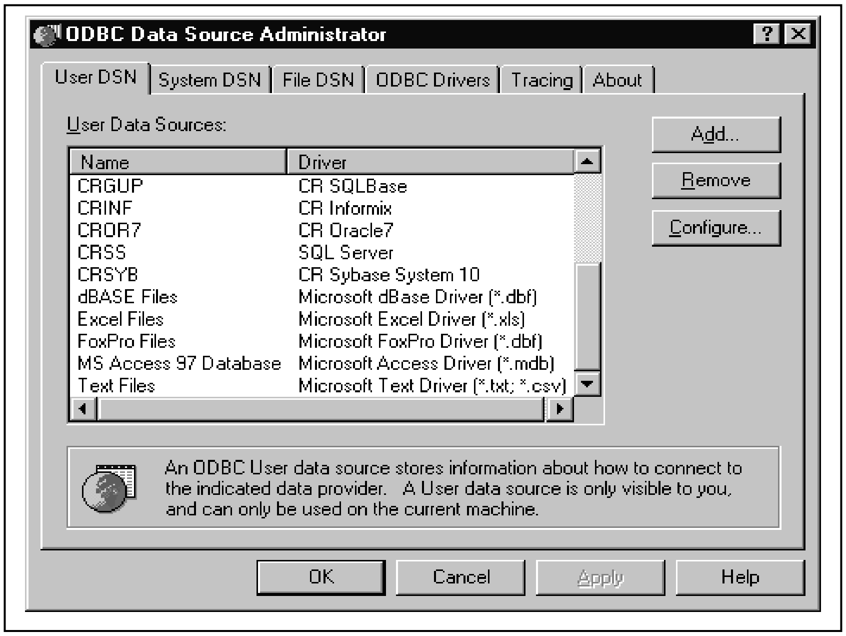
DESIGN OF DATABASE, INTERFACES AND SUPPORT MODULES

This section details the design and implementation of database tables, interfaces including the graphical user interface, and other support function modules that are required for the system to compute the delivery routes.

Data Extraction Interface

This chapter elaborates the setup of the interface for the Microsoft Access and development of queries for extracting the required data. The company has an ERP operating on an AS 400 system. The database is centralized and the system works on a mainframe computer. Since the routing system is a PC based system, the data needs to be extracted out of the AS 400 system to the Microsoft Access platform. For creating the interface, the Microsoft Windows 95 (and versions above) operating systems include the *Open Database Connectivity* (ODBC) setup utility. A Microsoft Access *Data Source Name* (DSN) needs to be created in this utility for the purpose of connecting to the company's database and to import data from the company's network. The Fig. 3 shows the screen print of the ODBC Data Source Administration form to set up the ODBC DSN for the required application.

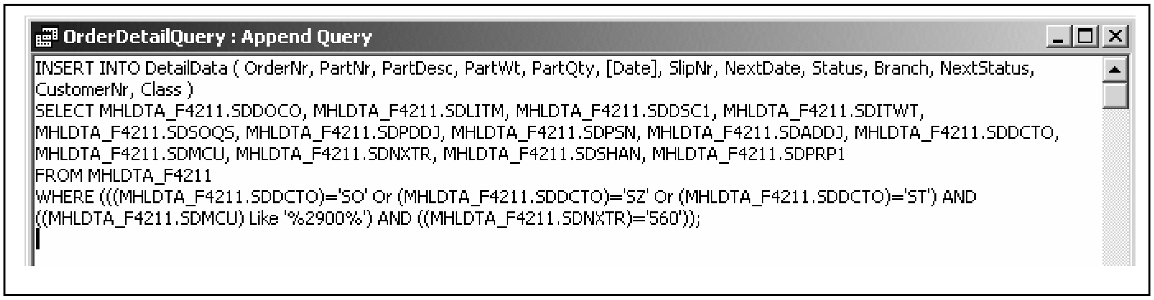
The next step is to import the database tables to the Microsoft Access. Tables containing the order information in the AS 400 system are imported to the routing system database. After importing, these data fields can be referred to in the program code for input/output manipulation using *data access object* (DAO) record-set declaration. The Structured Query Language (SQL) statement for order-detail extraction is given in Fig. 4.



Name	Driver
CRGUP	CR SQLBase
CRINF	CR Informix
CROR7	CR Oracle7
CRSS	SQL Server
CRSYB	CR Sybase System 10
dBASE Files	Microsoft dBase Driver (*.dbf)
Excel Files	Microsoft Excel Driver (*.xls)
FoxPro Files	Microsoft FoxPro Driver (*.dbf)
MS Access 97 Database	Microsoft Access Driver (*.mdb)
Text Files	Microsoft Text Driver (*.txt; *.csv)

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

Fig. 3. The ODBC setup screen of MS Windows 95



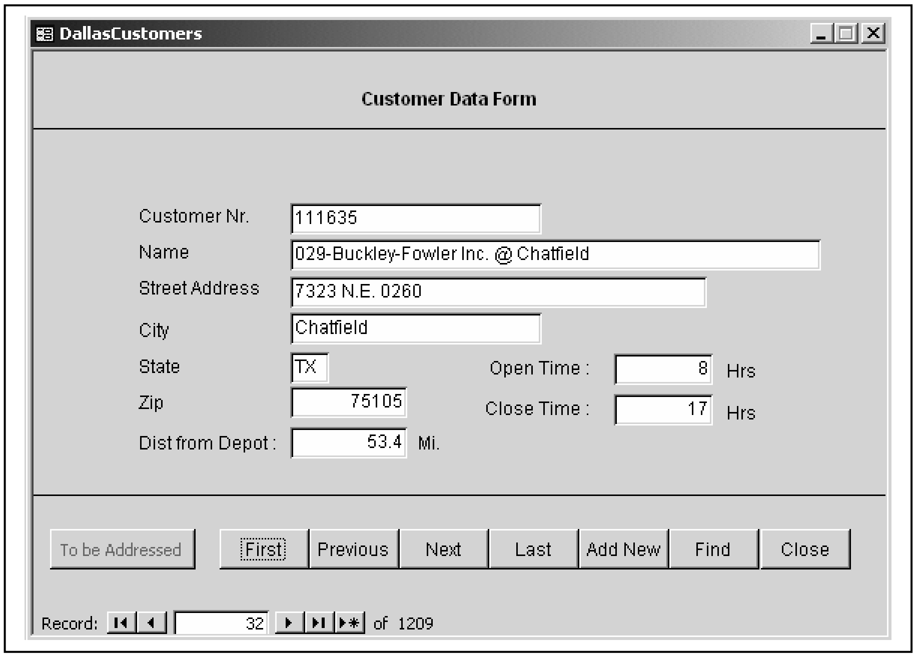
```
INSERT INTO DetailData ( OrderNr, PartNr, PartDesc, PartWt, PartQty, [Date], SlipNr, NextDate, Status, Branch, NextStatus,
CustomerNr, Class )
SELECT MHLDTA_F4211.SDDOCO, MHLDTA_F4211.SDLITM, MHLDTA_F4211.SDDSC1, MHLDTA_F4211.SDITWT,
MHLDTA_F4211.SDSOQS, MHLDTA_F4211.SDPDDJ, MHLDTA_F4211.SDPSN, MHLDTA_F4211.SDADDJ, MHLDTA_F4211.SDDCTO,
MHLDTA_F4211.SDMCU, MHLDTA_F4211.SDNXTR, MHLDTA_F4211.SDSHAN, MHLDTA_F4211.SDPRP1
FROM MHLDTA_F4211
WHERE (((MHLDTA_F4211.SDDCTO)='SO' Or (MHLDTA_F4211.SDDCTO)='SZ' Or (MHLDTA_F4211.SDDCTO)='ST') AND
((MHLDTA_F4211.SDMCU) Like '%2900%') AND ((MHLDTA_F4211.SDNXTR)='560'));
```

Fig. 4. A structured query language (SQL) statement for order extraction

Thus, the data extraction interface is designed and developed as a separate supporting module of the routing system to aid the querying of required data either by date or by order status.

Customer Database

The customer information such as customer number, name, street address, city, state and zip are stored in this database. The above-mentioned data are obtained from the AS 400 system for customers of the Dallas branch Regional Distribution Center (RDC). Then, their street level physical address for the corresponding delivery locations are compounded to form a parameter string, which when passed to the GIS interface along with the RDC's address, yields the distance between them. Fig. 5 shows the customer database form that is designed for the routing system.



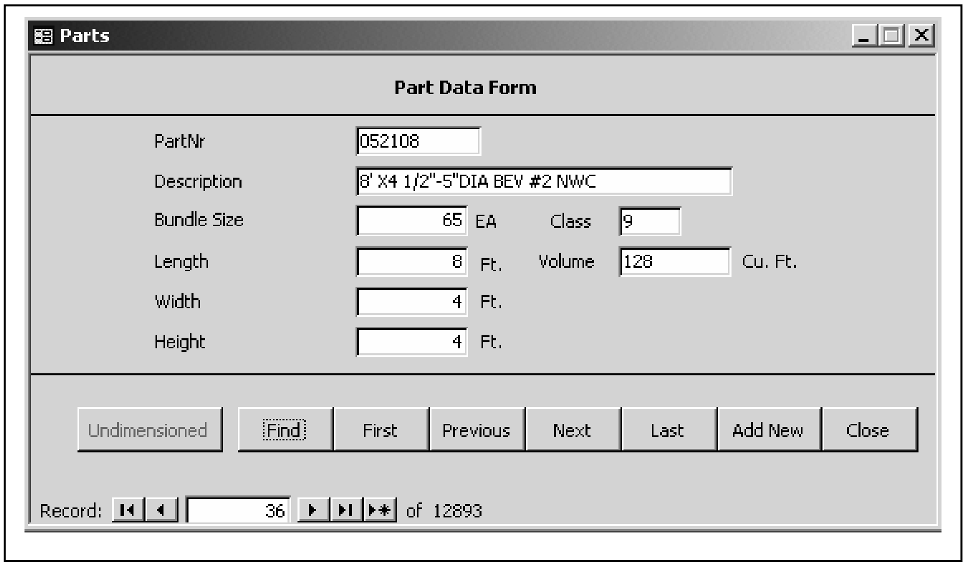
Customer Nr. 111635
Name 029-Buckley-Fowler Inc. @ Chatfield
Street Address 7323 N.E. 0260
City Chatfield
State TX
Zip 75105
Dist from Depot : 53.4 Mi.
Open Time : 8 Hrs
Close Time : 17 Hrs
Record: 32 of 1209

Fig. 5. Customer database form

Since a customer might place orders with different delivery locations, the user needs to select that delivery location where the customer currently needs it.

Parts Database

The parts database is implemented in a similar way as that of the customer database. The reason for having a separate part database is due to non-availability of the part specification such as part dimensions, weight, and bundle size and dimensions in the central database. The weight and dimension information are required for calculating the total truckload of parts and to check whether they exceed the truck capacity in any particular route. The database is deigned to have the columns of Part Nr, Part Description, Length, Breadth, Width, Bundle size and class as its fields. Fig. 6 shows the parts data screen.


Parts

Part Data Form

PartNr 052108
Description 8' X4 1/2"-5"DIA BEV #2 NWC
Bundle Size 65 EA Class 9
Length 8 Ft. Volume 128 Cu. Ft.
Width 4 Ft.
Height 4 Ft.

Record: 36 of 12893

Fig. 6. Parts database form

To develop the database the company provided the part specifications for certain part types. This forms the design of the parts database of the routing system.

Geographic Information System Interface

The routing algorithm requires the customer delivery location inter-distances for generating routes. For obtaining the distances, the Geographic Information System (GIS), the PC*MILER/STREETS 3.0 (PCMiler) (see reference) is used. A GIS is a computer system capable of assembling, storing, manipulating and displaying geographically referenced information, i.e., data identified according to their location. PCMiler is a GIS that can be used to calculate street level mileages for any origin-destination pair of locations, within the North American continent. Further, this GIS package has a graphic display namely, Route Map Window, which is like an electronic road atlas. Routes generated can be displayed on a map in the route map windows, enabling the examination of route details and visual evaluation of routing alternatives. A very important module of the PCMiler is the PC*MILER/STREETS Server (PCMiler Server). This can be used to calculate the mileage for an origin-destination pair of locations in hub mode. All the features of the PCMiler Server can be accessed from any development environment that is capable of handling Dynamic Link Library Functions (DLL). Fig. 7 shows a part of the VC++ code that is used to start the server, pass origin-destination data, get the mileage distance, and properly shut down the server.

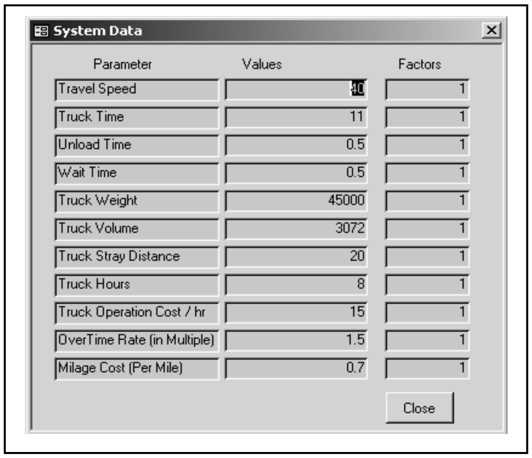
```
void UsePCMILER(){
PCMServerID server; server = PCMSOpenServer(0, 0);
/* Do other processing here. */
……..
/* Use the server: calculate trips, etc.... */
long PCMSCalcDistance (PCMServerID serv, const char FAR *orig, const char FAR *dest);
/* Shut down the server */
PCMSCloseServer(server);          }
```

Fig. 7. VC++ sample code for running PCMiler between two locations

The function PCMSOpenServer() will initialize the DLL, check the PCMiler licenses, load the PC*MILER/Streets highway database, and ready the engine for routing calculations. PCMSCalcDistance() returns the distance between *origin* and *destination* locations by calculating the route using the default routing type. Thus the PCMiler Server forms a basic interface tool for accessing the distance information from the routing system. This forms the design of the GIS interface module of the routing system.

Graphical User Interface

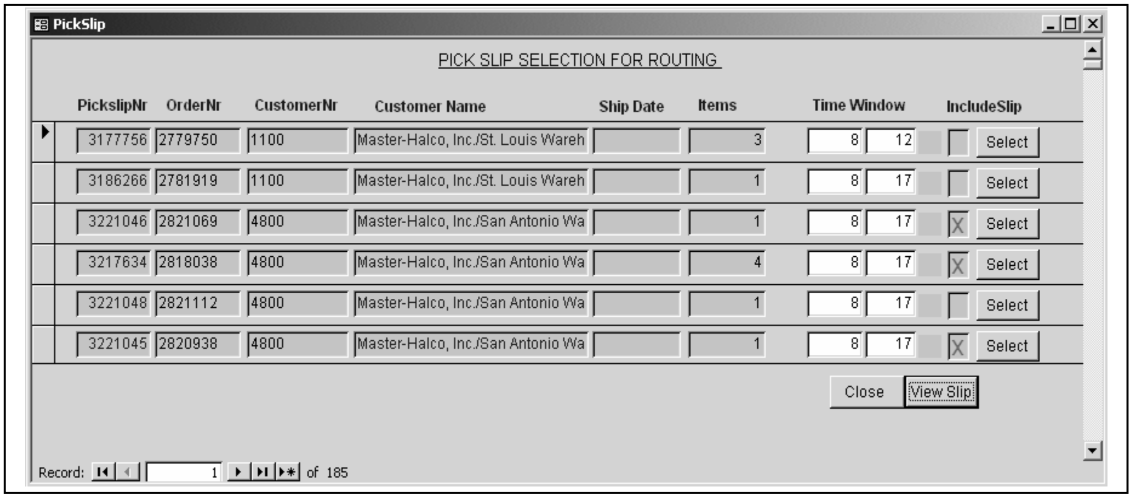
The Graphical User Interface is designed to provide user friendly, event-driven environment for the user to interact with the routing system. Modules and database of the routing system can be accessed through the main form named *Routing System Select Option*. This form is designed to have command buttons namely *System Data, Parts Data, Customers, Get Order Data, PickSlips, RunPCMiler, Get Distances, Route Orders, Print Routes, Load Orders,* and *Exit System* for triggering events or to open other forms. The System Data form shown in Fig 8, presents the data set from the parameter domain.



Parameter	Values	Factors
Travel Speed	40	1
Truck Time	11	1
Unload Time	0.5	1
Wait Time	0.5	1
Truck Weight	45000	1
Truck Volume	3072	1
Truck Stray Distance	20	1
Truck Hours	8	1
Truck Operation Cost / hr	15	1
OverTime Rate (in Multiple)	1.5	1
Milage Cost (Per Mile)	0.7	1

Fig. 8. System data form showing parameter domain data set

The customer data form is shown in Fig. 5 and the parts data form is shown in Fig. 6 are detailed in previous sub sections. The *Get Order Data* command button initiates the event for extracting customer order details from the mainframe system. A dial-up or a direct connection to the mainframe network needs to be established before the query is processed. Data extracted by this module is stored in corresponding tables within the routing system. The *pickslip* command button invokes the *SelectSlip* form, where the organized order information is presented for selection, as shown in Fig. 9. Here the details of the orders such as the pick slip number, customer name, time window details, and provision for including the slips for routing are available. The details of the customer orders can be obtained by clicking the *View Slip* command button that invokes the *Pickslip* form shown in Fig. 10. At this stage, the despatcher needs to verify the customer delivery addresses and corresponding time windows values. When the pickslips are finalized, the inter-distances check query is run by the *Run PCMiler* command button, to check the availability of inter-distance data between all the origin-destination combinations of the selected customer delivery locations, in the *LearnDistance* data table.

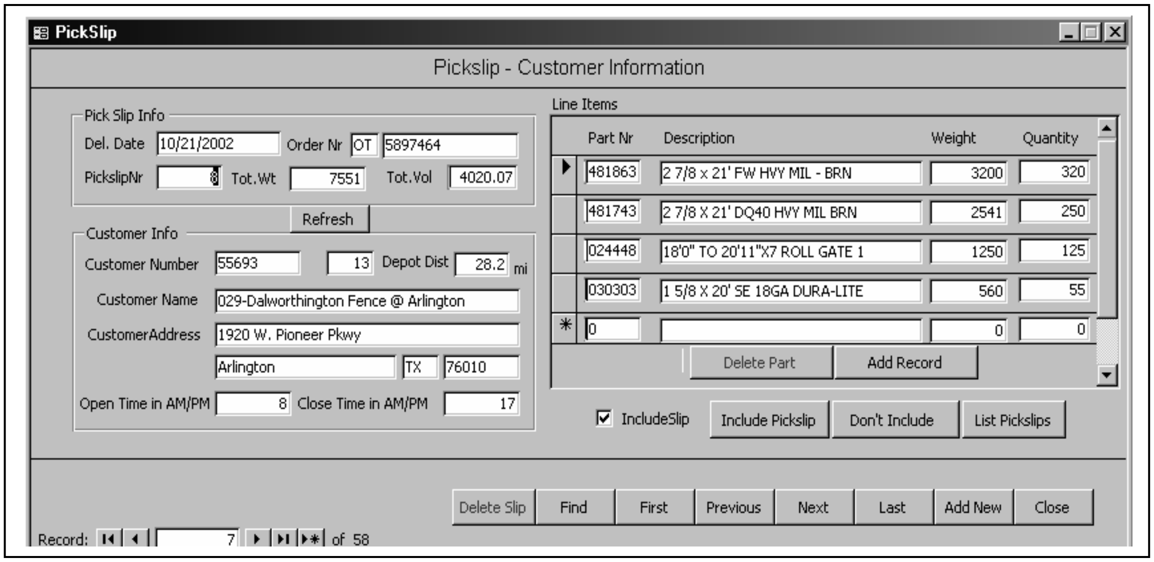


PICK SLIP SELECTION FOR ROUTING

PickslipNr	OrderNr	CustomerNr	Customer Name	Ship Date	Items	Time Window		IncludeSlip
3177756	2779750	1100	Master-Halco, Inc./St. Louis Wareh		3	8	12	☐
3186266	2781919	1100	Master-Halco, Inc./St. Louis Wareh		1	8	17	☐
3221046	2821069	4800	Master-Halco, Inc./San Antonio Wa		1	8	17	☒
3217634	2818038	4800	Master-Halco, Inc./San Antonio Wa		4	8	17	☒
3221048	2821112	4800	Master-Halco, Inc./San Antonio Wa		1	8	17	☐
3221045	2820938	4800	Master-Halco, Inc./San Antonio Wa		1	8	17	☒

Fig. 9. Pickslip selection form

After the GIS interface is executed and the distance generating process is complete, the *Get Distances* event should be executed to obtain and organize the inter-distance data into the transaction table namely *Distance*, which forms the input for the routing algorithm module.



Pickslip - Customer Information

Pick Slip Info

Del. Date 10/21/2002 Order Nr OT 5897464

PickslipNr 8 Tot.Wt 7551 Tot.Vol 4020.07

Customer Info

Customer Number 55693 13 Depot Dist 28.2 mi

Customer Name 029-Dalworthington Fence @ Arlington

CustomerAddress 1920 W. Pioneer Pkwy

Arlington TX 76010

Open Time in AM/PM 8 Close Time in AM/PM 17

Line Items

Part Nr	Description	Weight	Quantity
481863	2 7/8 x 21' FW HVY MIL - BRN	3200	320
481743	2 7/8 X 21' DQ40 HVY MIL BRN	2541	250
024448	18'0" TO 20'11"X7 ROLL GATE 1	1250	125
030303	1 5/8 X 20' SE 18GA DURA-LITE	560	55
0		0	0

☑ IncludeSlip

Record: 7 of 58

Fig. 10. Pickslip detail form

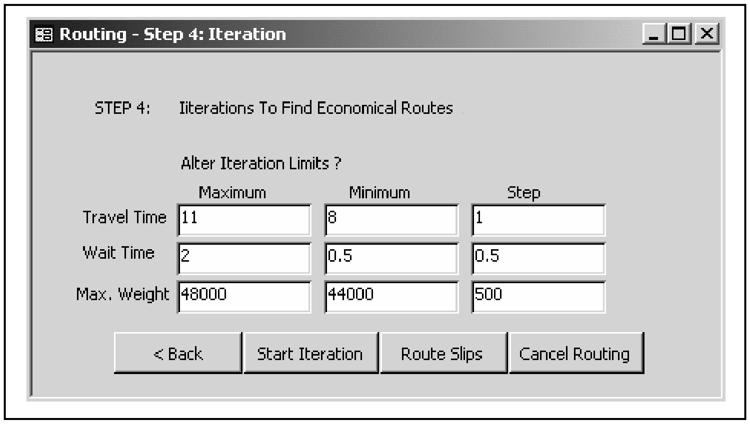
The Route Order command button invokes a five step routing process, similar to a wizard. These steps includes,

Step 1: Confirm the pick-slip selection, thereby confirming the delivery location addresses and the time window values.

Step 2: Modify the parameter domain data.

Step 3: Do a what-if analysis or proceed to step 5.

Step 4: What-if analysis can be done by providing appropriate values for the parameters and the step size in the text boxes as shown in Fig. 11. Following the iteration, the results appear.



STEP 4: Iterations To Find Economical Routes

Alter Iteration Limits ?

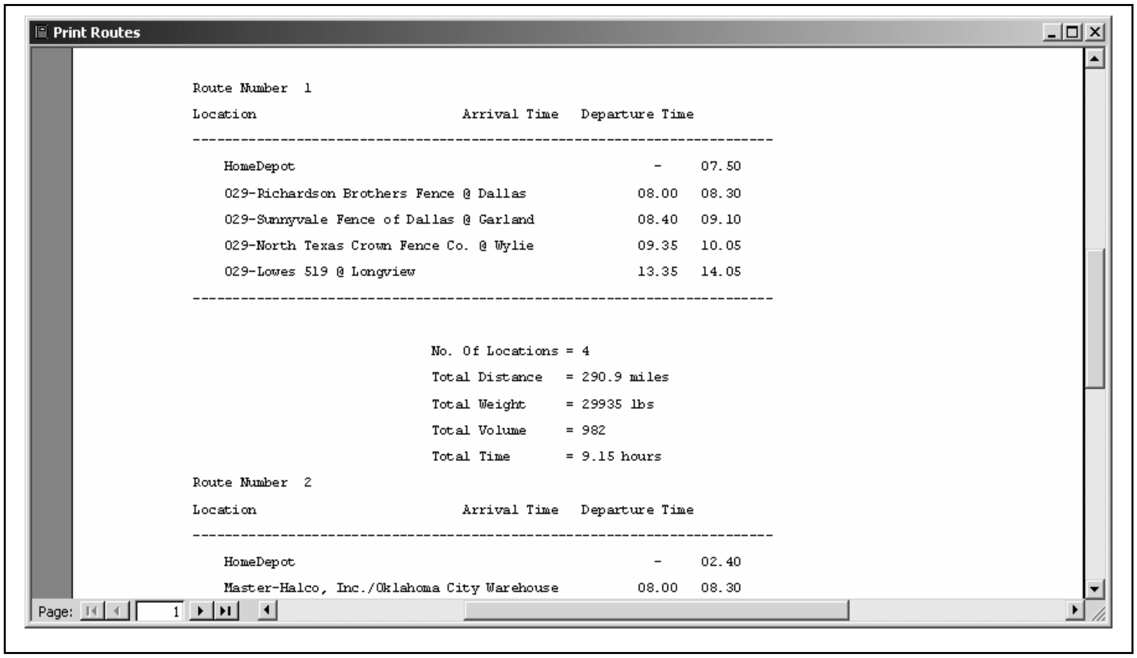
	Maximum	Minimum	Step
Travel Time	11	8	1
Wait Time	2	0.5	0.5
Max. Weight	48000	44000	500

Fig. 11. Iteration step to perform what-if analysis

Step 5: this step invokes the routing algorithm module and presents the generated routes in a tree structure. From here on, the additional side constraints can be added for re-optimization. Thus, the route order procedure incorporates the routing algorithm, re-optimization and what-if analysis modules.

The Print Routes command invokes a report as shown in Fig. 12, which is designed to the requirements of the company. The format includes the route number, customer name, arrival and departure timetables, and route schedule summary along with costs of operation. Further, the routes can be exported to the PCMiler and can be viewed as shown in Fig 13.

The *load order* exports data required for loading/cubing system. The *Exit System* shuts down the routing system. Thus a user friendly Graphical User Interface (GUI) is designed to provide an event-driven environment for the user to interact with the routing system.



```
Route Number  1
Location                          Arrival Time   Departure Time
-------------------------------------------------------------------------
    HomeDepot                                           -      07.50
    029-Richardson Brothers Fence @ Dallas            08.00    08.30
    029-Sunnyvale Fence of Dallas @ Garland           08.40    09.10
    029-North Texas Crown Fence Co. @ Wylie           09.35    10.05
    029-Lowes 519 @ Longview                          13.35    14.05
-------------------------------------------------------------------------

                              No. Of Locations = 4
                              Total Distance   = 290.9 miles
                              Total Weight     = 29935 lbs
                              Total Volume     = 982
                              Total Time       = 9.15 hours
Route Number  2
Location                          Arrival Time   Departure Time
-------------------------------------------------------------------------
    HomeDepot                                           -      02.40
    Master-Halco, Inc./Oklahoma City Warehouse        08.00    08.30
```

Fig. 12. Print report layout

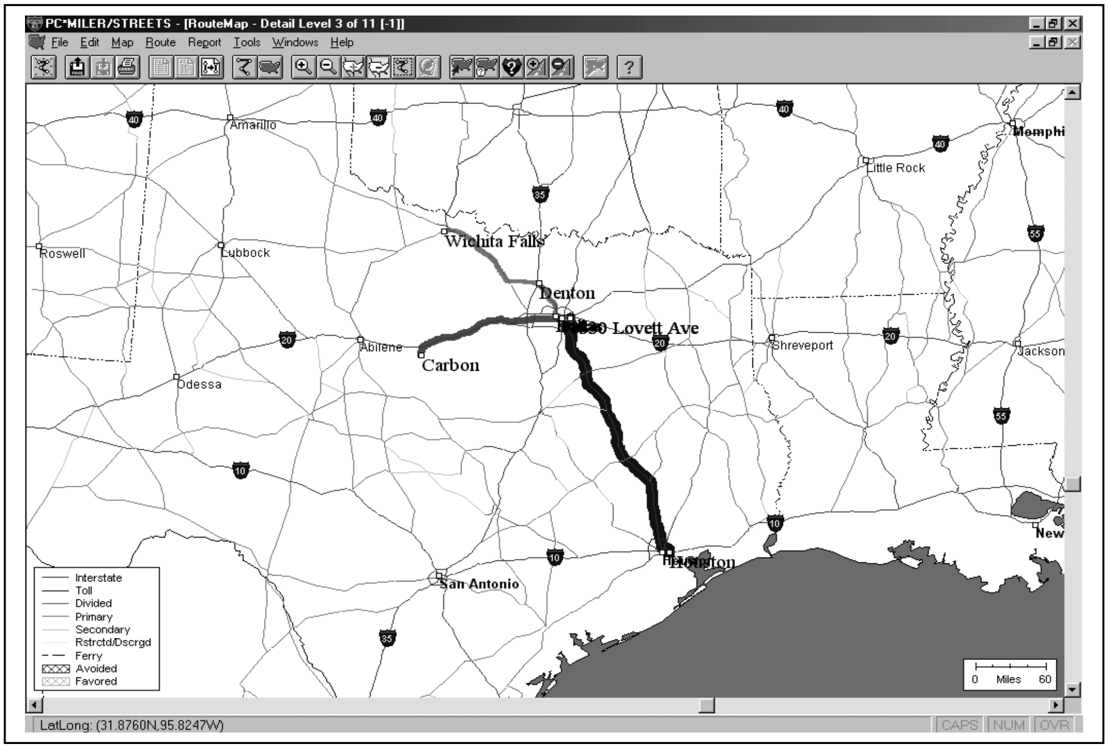


Fig. 13. Visual layout of routes as seen in PCMiler Streets

Inter-Distance Learning/Maintenance Module

The inter-distance learning module is a part of the maintenance processing, which influences the performance of the system. This module needs periodic maintenance by a Systems Engineer or System Administrator.

The customer inter-distances that form the interface domain data, is very much crucial to the routing algorithm. The GIS interface module is used for obtaining the distance data for an origin-destination pair. Since the process involves communication through PC*MILER STREETS 3.0/Server and the information requires physical addresses at street level, it takes considerable time to get data for a single pair. Average time taken for such communication is observed to be 15 seconds per origin-destination pair. The customer inter-distance learning / maintenance module is designed to achieve this by implementation of a database table and storing the origin-destination pairs along with their distances. This database table contains the data for those customers who place order very frequently with the company.

Based on the history file data available at the company end, it is observed that about 250 customers have placed orders frequently, with the Dallas branch RDC of the company. So, the inter-distance data for these customers are generated at one-shot and the data is stored in the *LearnDistance* table. Now, before executing the routing algorithm, the inter-distances for those customers for whom the data is not available in the *LearnDistance* table needs to be obtained through the PCMiler. In experience of carrying out routing for about a couple of weeks, as a study, it was observed that utmost three to four new customer delivery locations or changes in the street address of the existing customers occurred. Thus, assuming symmetry of distance between an origin-destination pair, the number of transactions previously required is given by $[(N-1)/2 * N + N]$, where N is the number of customer delivery locations, the two components of the expression being the customer inter-distances and the depot distances to each customer delivery location. Now, assuming M to be the new set of customers, for whom the data are not available in the *LearnDistance* table, we need to communicate with the PCMiler Server just to obtain [(N-M)*M+M+(M-1)/2*M], the three terms each representing the inter-distances among the new set and the existing set of customers, the depot distances for the new set of customers, and the inter-distances among the new set of customers.

The maintenance sub-module maintains the *LearnDistance* database table, by appending or removing origin-destination pairs along with their inter-distance, based on the history data. After routing is done, the customer order (pick-slip number and customer number) data are stored in a history database maintained separately as *PickslipInfo.mdb*. This database is designed exclusively for holding this history data and it grows with the time. When maintenance is performed, a query is run through the history database to find the number of hits per customer for a period of time. Based on the number of hits per customer, a critical value is determined (either input by Systems Engineer or default of 12 hits per year), those customer locations not available in the *LearnDistance* database table are appended to the it after obtaining their inter-distances. The number of origin-destination pairs for which distances needs to be obtained, assuming symmetry is given by [(N1*M1) + M1 + (M1-1)/2 * M1)], where N1 is the total number of customers available in the *LearnDistance* database and M1 is the number of customer locations to be appended to the table. This process takes considerable time, and requires considerable PCMiler Server time.

Maintenance can also be performed to remove origin-destination pairs from the *LearnDistance* table. This aids in reducing the query processing time at the time of routing, when the inter-distances are required. Thus maintenance processing is carried out in the routing system. The following section reviews the modeling approaches and solution methods available to solve vehicle routing problems.

LITERATURE REVIEW FOR ALGORITHM DESIGN

The vehicle routing problem (VRP) concerns the identification of minimal cost routes for a given set of delivery locations. Since vehicle capacities are limited, restrictions can be imposed on the total weight and total volume that a vehicle can carry in a route. This defines the capacitated vehicle routing problem (CVRP). An important extension of CVRP is the vehicle routing problem with time-windows (CVRPTW). A significant amount of research has been done for solving CVRPTW and its variants. For surveys in this area see Solomon and Desrosiers, 1988; Bramel and Simchi-Levi, 1997. The VRPTW is classified under the class of strongly NP-Hard problem Savelsberg, 1985. Hence, the solution time to reach an optimal cost routes might take exponential time as the number of customer locations. The following subsection gives the network based modeling of CVRPTW.

Network Formulation of CVRPTW

Cordeau et al., 2002 formulates VRPTW on a network G = (N, A), where the depot is represented by two nodes "*o*" and "*d*". All feasible routes correspond to paths from node *o* to node *d*. With each arc (i, j) in A, where $i \neq j$, a cost c_{ij} and a time t_{ij} (including the service time at node i) are defined. Each vehicle has a capacity Q and each customer has a demand q_i. A time window denoted by $[a_i, b_i]$ is associated with customer i. The model contains two sets of decision variables namely x and s. For each arc (i; j), where $i \neq j$; $i \neq n + 1$; $j \neq 0$, and each vehicle k we define x_{ijk} as

$$x_{ijk} = \begin{cases} 0, \text{ if vehicle k doesnot include (i,j)} \\ 1, \text{ if vehicle k includes (i,j) in its path.} \end{cases}$$

The decision variable s_{ik} is defined for each vertex i and each vehicle k and denotes the time vehicle k starts to service customer i. In case the given vehicle k does not service customer i, s_{ik} does not mean anything. Minimal cost routes one for each vehicle belonging to set V needs to be designed such that each customer is serviced exactly once, every route originates at vertex *o* and ends at vertex *d*, and the time windows and capacity constraints are observed. The VRPTW can be mathematically formulated as shown in Fig 14 (derived from Cordeau et al., 2002).

$$\text{Minimize } z_{V\,RPTW} = \sum_{k\in V}\sum_{i\in N}\sum_{j\in N} c_{ij}x_{ijk} \tag{1}$$

subject to

$$\sum_{k\in V}\sum_{j\in N} x_{ijk} = 1 \qquad \forall\, i\in N\setminus\{o,d\} \tag{2}$$

$$\sum_{i\in N/\{o,d\}} q_i \sum_{j\in N} x_{ijk} \le Q \qquad \forall\, k\in V \tag{3}$$

$$\sum_{j\in N} x_{ojk} = 1 \qquad \forall\, k\in V \tag{4}$$

$$\sum_{i\in N} x_{ihk} - \sum_{j\in N} x_{hjk} = 0 \qquad \forall h\in N\setminus\{o,d\},\ \forall\, k\in V \tag{5}$$

$$\sum_{j\in N} x_{jdk} = 1 \qquad \forall\, k\in V \tag{6}$$

$$s_{ik} + t_{ij} - M(1- x_{ijk}) \le s_{jk} \qquad \forall i,j\in N,\ \forall\, k\in V \tag{7}$$

$$a_i \le s_{ik} \le b_i \qquad \forall\, i\in N,\ \forall\, k\in V \tag{8}$$

$$x_{ijk} \in \{0,1\} \qquad \forall\, i,j\in N,\ \forall\, k\in V \tag{9}$$

$$s_{ik} \ge 0 \qquad \forall i\in N,\ \forall\, k\in V \tag{10}$$

Fig. 14. Mathematical formulation of CVRPTW

The objective function (1) is to minimize the cost associated with the decision variable xijk. Constraint (2) is to allow a customer delivery in one and only one vehicle. Constraint (3) requires that the sum of the weights of demand in every route should not exceed the capacity of the vehicle. Constraints (4)-(6) are flow conservation constraints for each vehicle. Constraint (7), (9) and (10) represents time window constraints, and constraint on arrival time s_{ik} at node i by vehicle k.

Solution techniques available can be classified into different categories such as heuristic techniques, meta-heuristics, optimization based heuristics, and optimization algorithms as detailed in Cordeau et al., 2002. Next, we discuss each category.

Local Search Heuristics

Heuristic methods perform a relatively limited exploration of the solution search space and typically produce good quality solutions within modest computing times. Bodin and Golden, 1981, provides an overview of approaches to the CVRP. Following the survey presented in Christofides et al., 1978, the heuristics for solving CVRP can be classified as follows.

Cluster first – route second heuristics

This procedure clusters the customer locations into routes based on the capacity restrictions on the vehicle, without regard for any sequence of delivery. These routes, called feasible routes are then sorted to obtain economical routes. The two-phase method (Christofides et al., 1978), the generalized assignment heuristic (Fisher and Jaikumar, 1981), and the location based heuristic (Bramel and Simchi-Levi, 1995) can be classified under this category.

Route first – cluster second heuristics

First, a large route is constructed which includes all of the customer locations. Next, this large route is partitioned into a number of smaller and feasible routes (based on the capacity restrictions). One vehicle is assigned to each segment and the sequence of delivery may be altered to obtained economical routes. The optimal partitioning heuristic (Beasley, 1983), and the sweep algorithm (Gillett and Miller, 1974) can be categorized under this class of heuristics.

Savings/insertion heuristics

The savings algorithm (Clarke and Wright, 1964) is one of the earliest heuristics. This approach starts with an initial solution that assigns each customer to a separate vehicle. Then, at each step of the heuristic, a current configuration is compared to an alternative configuration obtained by combining two routes without violating capacity restrictions. The configuration that yields largest savings in terms of cost function is chosen for next iteration. The procedure eventually concludes with feasible and economical routes.

The above mentioned heuristics are well established in literature to solve specific problem instances at a lesser time. However, these procedures do not guarantee a global optimal solution.

Meta-Heuristics

Unlike heuristic methods, which are local search methods that terminate once a local optimum is reached, these methods are aimed at exploring a larger subset of the solution space in the hope of finding a near-optimal solution. Tabu search, simulated annealing, evolutionary algorithms, and ant colony optimization are few methods that can be classified under meta-heuristic approaches. Carlton, 1995; Potvin et al, 1996 uses tabu search technique to solve the CVRPTW. Chiang and Russell, 1996, describes a simulated annealing meta-heuristics to solve CVRPTW. Solution techniques using genetic algorithms has been devised by Potvin and Bengio, 1996; Blanton and Wainwright, 1993. Gambardella et al, 1999 uses the ant colony optimization technique to solve CVRPTW.

Optimization Based Heuristics

Approximation methods can also be derived directly from optimization algorithms, by heuristically solving different phases of the process. Partial exploration of branch-and-bound tree is another such technique that can be used to find an acceptable solution based on time availability. An integer solution can be obtained by using a depth-first strategy. Now, the tree can be explored for the remaining available CPU time. Alternatively, elimination of branches on heuristic ground rules accelerate the decision process and may provide quite good solutions. Koskosidis et al, 1992 uses a mixed integer programming model to generalize the Fisher and Jaikumar, 1981 heuristic for CVRPTW.

Optimization Algorithms

This approach proposes to compute every possible solution until one of the best is reached. Brach-and-bound search method as described by Fisher, 1994, Lagrangian Dual Method as given in Kallehauge, 2001, and Branch-and-price as described in Desrochers et al, 1992 are few of the exact optimization algorithms. Although these methods guarantee and optimal solution at the end of the search, the computational time required may be large.

ALGORITHM DESIGN FOR CVRPTW WITH RUSH-ORDER PROCESSING

This module is the heart of the routing system that performs the primary function of generating cost effective routes based on the weight and volume capacity of the fleet, time-window constraints on the order delivery, and changes due to rush-order arrivals. The following subsection gives the formulation and the solution methodology for the problem, based on the literature review in the previous section.

Formulation of the Mathematical Model

The mathematical formulation is described by defining necessary notation and introducing constraint equations and objective function terms which constitutes the mixed linear integer programming problem.

Network and sets

Let $G = (N, A)$ be a network with a set of N nodes and a set of A arcs. The 0^{th} node is the company's distribution center. Set V consists of K types of vehicles having a m_k number of trucks of type k. Set K consists of identical fleet from 1,2, …m_k. Set P consists of maximum number of delivery locations that can be accommodated in a route.

Indices

$i, j = 1, 2, 3\ldots n$ index the customer delivery location nodes in the network $G = (N,A)$ (where i is a origin node and j is the destination node). If stated $i \varepsilon N$, then i includes node 0.

$l = 1, 2, 3\ldots m_k$ indexes vehicles of type k.

$p = 1, 2, 3\ldots$ P represents the order and position of a node in a delivery sequence.

$r = 1, 2, 3 \ldots R$ indexes the number of different part types in a customer order.

Decision variables

X_{ijlp} – equals 1 if arc $(i,j)\ \varepsilon\ A$ is used by the vehicle l in position p; 0 otherwise.

T_{il} – arrival time at node i by vehicle l.

OT_l – overtime assignment for vehicle l.

Input parameters

Qw_k –weight capacity for the vehicle (lbs) type k.

Qv_k –volume capacity for the vehicle (cu. ft) type k.

e – maximum permissible time including the over time (min).

e' – regular time of any particular route taken (min).

ao_l – open time of the time window of i_{th} customer (min).

a_i – open time of the time window of i_{th} customer (min).

b_i – open time of the time window of i_{th} customer (min).

Pw_{ir} – total weight of r_{th} part type in pickslip order of i_{th} customer (lbs).

Pv_{ir} – total volume of r_{th} part type in pickslip order of i_{th} customer (cu. ft).

nr_{ir} – quantity of r_{th} part type in the pickslip order of i_{th} customer.

bs_r – bundle size of *rth* part type (cu. ft).

d_{ij} – actual mileage between the customer delivery locations represented by nodes i and j, if a commercial heavy vehicle were to travel between the locations (mi).

C_m – mileage cost in cost per unit mile.

C_f – fixed cost per hour incurred, if a vehicle were to be routed.

C_o – overtime costs. This is 1.5 times the fixed cost.

VS – average travel speed of the fleet (mph).

U_i – average time to unload customer i's order at a delivery site (min).

RPC_{ilp} – repositioning cost for node i within a same truck due to sequence change.

Given an initial set of routes, RPC_{ilp} is a vector of cost values with non-zero entries corresponding to the entries $i' = i$, $l'=l$, and $p' \neq p$, $\forall\ X_{ijlp} = 1$.

RTC_{il} – reloading cost for node i when unloaded from truck k due to sequence change.

Given an initial set of routes, RTC_{il} is a vector of cost values with non-zero entries corresponding to the entries $i' = i$, and $l' \neq l$, $\forall\ X_{ijlp} = 1$.

δ – very small number in the order of x 10^{-3} used as a dummy cost parameter.

Derived inputs

w_i – gives the total weight of all parts of the i_{th} customer order.

$$\text{where } w_i = \sum_{r=1}^{R} Pw_{ir} \,.\, nr_{ir}$$

v_i – gives the total volume of all parts of the i_{th} customer order.

$$\text{where } v_i = \sum_{r=1}^{R} (Pv_{ir} \,.\, nr_{ir}) / bs_r$$

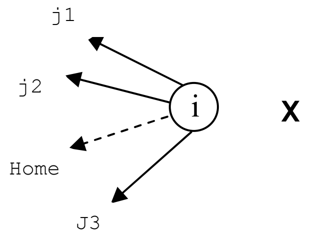
t_{ij} – gives the time taken to travel from node i to node j

$$\text{where } \quad t_{ij} = d_{ij} / VS$$

Constraints and objective function

1. Constraints on network flow: Each customer delivery should to be made once in a route. This can be modeled by restricting only one arc to originate and terminate at a node. These conditions are given by equations (11) and (12) as shown in Fig. 15.

$$\Sigma_{j \in N}\, \Sigma_{l \in K \in V}\, \Sigma_{p \in P}\ X_{ijlp} \qquad = 1 \qquad \forall i \varepsilon N \backslash \{o\} \qquad (11)$$



$$\Sigma_{i \in N}\, \Sigma_{l \in K \in V}\, \Sigma_{p \in P}\ X_{ijlp} \qquad = 1 \qquad \forall j \varepsilon N \backslash \{o\} \qquad (12)$$

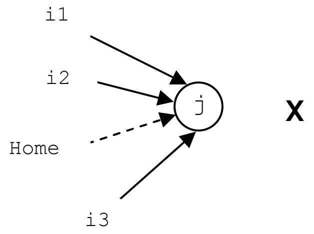
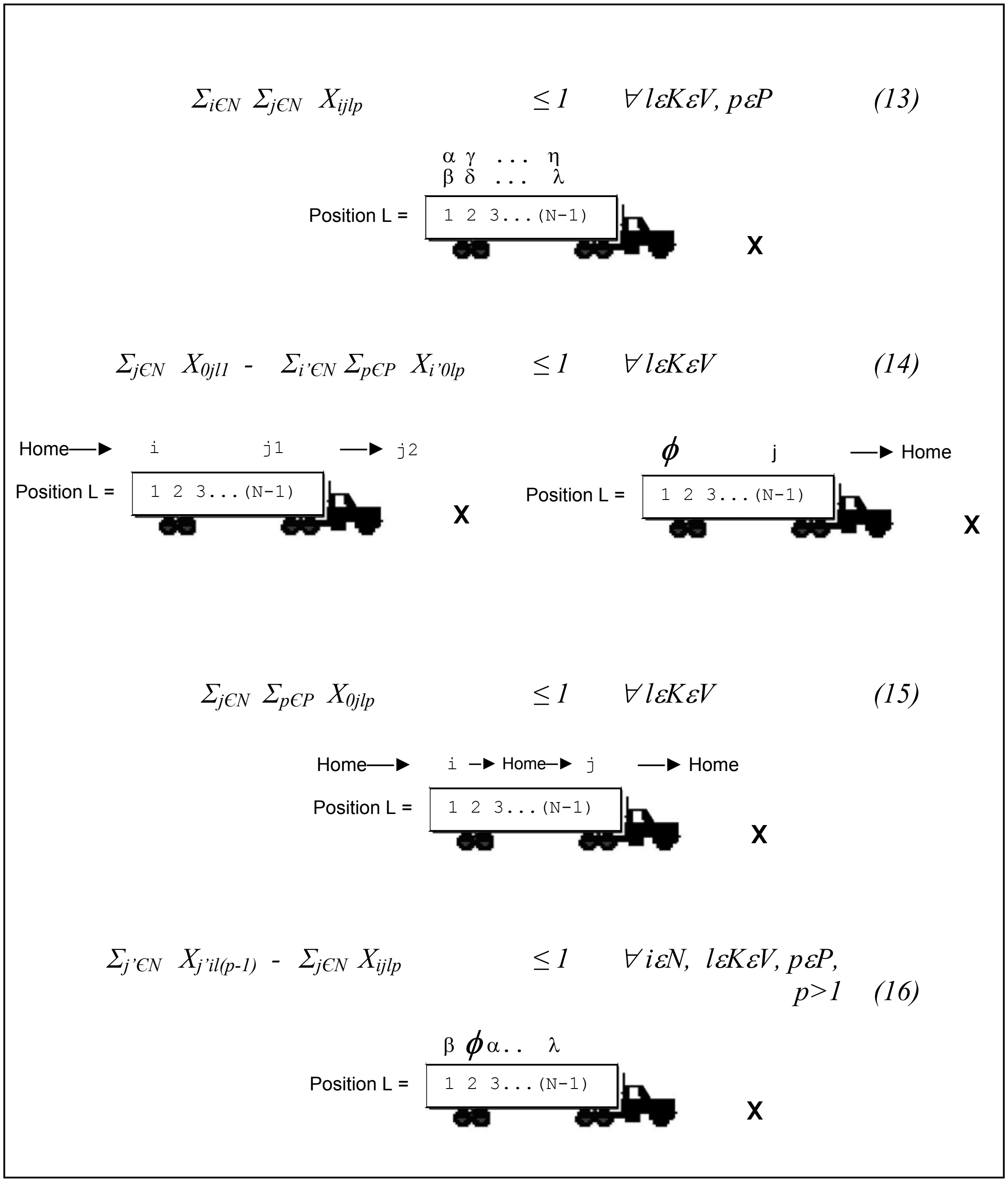


Fig. 15. Network flow constraints

2. Constraints on delivery sequence and positions: The company has only m_k vehicles in type k and K such types available represented by set V for routing and each vehicle can deliver only one customer at a time, as given in (13) of Fig 16. Further, the vehicle should start from the distribution center and complete its delivery and return to the distribution center. This is achieved by constraints (14) in Fig 16. Also, only one trip per vehicle from the RDC with no empty position within a delivery route is permitted. These constraints (15) and (16) are formulated as shown in Fig 16.



$$\Sigma_{i\in N}\ \Sigma_{j\in N}\ X_{ijlp} \le 1 \qquad \forall l\varepsilon K\varepsilon V,\ p\varepsilon P \qquad (13)$$

$$\Sigma_{j\in N}\ X_{0jl1} - \Sigma_{i'\in N}\ \Sigma_{p\in P}\ X_{i'0lp} \le 1 \qquad \forall l\varepsilon K\varepsilon V \qquad (14)$$

$$\Sigma_{j\in N}\ \Sigma_{p\in P}\ X_{0jlp} \le 1 \qquad \forall l\varepsilon K\varepsilon V \qquad (15)$$

$$\Sigma_{j'\in N}\ X_{j'il(p-1)} - \Sigma_{j\in N}\ X_{ijlp} \le 1 \qquad \forall i\varepsilon N,\ l\varepsilon K\varepsilon V,\ p\varepsilon P,\ p>1 \qquad (16)$$

Fig. 16. Constraints on delivery sequence and positions

3. Constraints on weight and volume capacities: The sum total of part weights of every customer in any of the routes should not exceed the maximum weight capacity specification (safe load

capacity) of trucks. Similar to weight, the sum total of part volume of every customer in any of the routes should not exceed the maximum volume capacity specification (safe load capacity) of trucks. These are satisfied by the capacity constraints (17) and (18) shown in Fig 17.

$$\Sigma_{i\in N} \Sigma_{j\in N} \Sigma_{p\in P} w_i X_{ijlp} \leq Qw_k \qquad \forall l\varepsilon K\varepsilon V \qquad (17)$$

$$\Sigma_{i\in N} \Sigma_{j\in N} \Sigma_{p\in P} v_i X_{ijlp} \leq Qv_k \qquad \forall l\varepsilon K\varepsilon V \qquad (18)$$

Fig. 17. Constraints on weight and volume capacities

4. Constraints on time-windows and time additivity: At any customer delivery location, the time of arrival of a truck (T_{jl}) is the time taken to travel from node $i\varepsilon N$ to node $j\varepsilon N$ given by t_{ij}, plus the time of arrival at node i (T_{il}) and the unloading time U_i as given in (20) of Fig 18. For each truck l, there is a start time ao_l. Hence, for a customer to be first visited by a truck, the T_{il} should be replaced by ao_l as in (19) of Fig 18. The time window constraints are formulated by restricting the value of T_{il} for customer i and truck l within the interval $[a_i, b_i]$ as given by (21) and (22) of Fig 18. For every trip, the total time including the over time, as set forth by State/Federal and company regulations, should not exceed the total time limit of e. The regular duration of a delivery route is given by e'. The overtime duration for each truck needs to be calculated to incorporate the overtime cost term in the objective function. This can be found by considering the difference between the time of arrival to the RDC and the regular time duration e'. The overtime duration should not exceed the difference between the maximum permissible time e and the regular time e'. These constraints (23) and (24) are formulated as in Fig 18.

$$T_{il} - (ao_l + t_{oi})\, \Sigma_{p\in P} X_{0ilp} \geq 0 \quad \forall i\varepsilon N,\ l\varepsilon K\varepsilon V \quad (19)$$

$$T_{jl} - T_{il} - (U_i + t_{ij})\, \Sigma_{p\in P} X_{ijlp} \geq 0 \quad \forall i\varepsilon N,\ l\varepsilon K\varepsilon V \quad (20)$$

$$T_{il} - a_i\, \Sigma_{j\in N} \Sigma_{p\in P} X_{ijlp} \geq 0 \quad \forall i\varepsilon N,\ l\varepsilon K\varepsilon V \quad (21)$$

$$T_{il} - b_i\, \Sigma_{j\in N} \Sigma_{p\in P} X_{ijlp} \leq 0 \quad \forall i\varepsilon N,\ l\varepsilon K\varepsilon V \quad (22)$$

$$OT_l - T_{il} + e' \geq 0 \quad \forall i\varepsilon N,\ l\varepsilon K\varepsilon V \quad (23)$$

$$OT_l - e + e' \leq 0 \quad \forall l\varepsilon K\varepsilon V \quad (24)$$

Fig. 18. Constraints on time-windows and time additivity

5. Restrictions on decision variables: All decision variables are required to satisfy non-negativity constraints. Further, the decision variable X_{ijlp} needs to satisfy binary restrictions. These are given by (25), (26), and (27) in Fig 19.

$$X_{ijlp} \quad \varepsilon \{0,1\} \quad \forall i\varepsilon N,\ j\varepsilon N,\ l\varepsilon K\varepsilon V,\ p\varepsilon P \quad (25)$$

$$T_{il} \quad \geq 0 \quad \forall i\varepsilon N,\ l\varepsilon K\varepsilon V \quad (26)$$

$$OT_l \quad \geq 0 \quad \forall l\varepsilon K\varepsilon V \quad (27)$$

Fig. 19. Restrictions on decision variables

6. Objective function: The objective of the CVRPTW with rush order processing is to minimize the total cost of routing. The cost components includes the mileage cost, overtime costs,

reloading costs if customer orders were to be unloaded and reloaded. Reloading costs can be of two types. It can be either the cost associated with repositioning the order within the same route, or unloading and reloading the order to a different route altogether. The objective function is modeled as shown in (28) of Fig 20. The objective function also includes a dummy cost for adjusting the T_{il} values to its minimum.

$$\begin{aligned} \text{Minimize } Z_{VRPTWRO} = {} & C_m * \Sigma_{i\in N} \Sigma_{j\in N} \Sigma_{l\in K\in V} \Sigma_{p\in P}\ d_{ij} * X_{ijlp} + C_o * \Sigma_{l\in K\in V}\ OT_l \\ & + \Sigma_{i\in N} \Sigma_{l\in K\in V} \Sigma_{p\in P}\ RPC_{ilp} * (1 - \Sigma_{j\in N} X_{ijlp}) \\ & + \Sigma_{i\in N} \Sigma_{l\in K\in V}\ RTC_{il} * (1 - \Sigma_{j\in N} \Sigma_{p\in P} X_{ijlp}) \end{aligned} \tag{28}$$

Fig. 20. Objective function for CVRPTW with rush-order processing

This formulation, shown in Fig. 21, is coded using AMPL-CPLEX to solve for the optimal cost routes. The formulation is tested using the company data. The run times are very large that even for small instances of the problem, the problem takes an exponential time (results are presented in the following section) and this verifies the difficulty of solving the problem as mentioned in Savelsberg, 1985. Hence, heuristics based on business rules that could solve the CVRPTWRO at a very short time for a feasible and reasonably good solution are developed. A short computational time is important because it is not practicable for the logistics department of the company to wait for an indeterminate time to make vehicle assignments. Hence, any approach involving enumeration techniques will take exponential time that depends on the number of customers, width of the time-windows, and number of vehicles. The following section briefs the heuristics implemented in the routing system.

$$\begin{aligned} \text{Minimize } Z_{VRPTWRO} = {} & C_m * \Sigma_{i\in N} \Sigma_{j\in N} \Sigma_{l\in K\in V} \Sigma_{p\in P}\, d_{ij} * X_{ijlp} + C_o * \Sigma_{l\in K\in V}\, OT_l \\ & + \Sigma_{i\in N} \Sigma_{l\in K\in V} \Sigma_{p\in P}\, RPC_{ilp} * (1 - \Sigma_{j\in N} X_{ijlp}) \\ & + \Sigma_{i\in N} \Sigma_{l\in K\in V}\, RTC_{il} * (1 - \Sigma_{j\in N} \Sigma_{p\in P} X_{ijlp}) \end{aligned} \tag{28}$$

$$\Sigma_{j\in N} \Sigma_{l\in K\in V} \Sigma_{p\in P}\, X_{ijlp} = 1 \qquad \forall\, i \varepsilon N\backslash\{o\} \tag{11}$$

$$\Sigma_{i\in N} \Sigma_{l\in K\in V} \Sigma_{p\in P}\, X_{ijlp} = 1 \qquad \forall\, j \varepsilon N\backslash\{o\} \tag{12}$$

$$\Sigma_{i\in N} \Sigma_{j\in N}\, X_{ijlp} \leq 1 \qquad \forall\, l\varepsilon K\varepsilon V,\, p\varepsilon P \tag{13}$$

$$\Sigma_{j\in N}\, X_{0jl1} - \Sigma_{i'\in N} \Sigma_{p\in P}\, X_{i'0lp} \leq 1 \qquad \forall\, l\varepsilon K\varepsilon V \tag{14}$$

$$\Sigma_{j\in N} \Sigma_{p\in P}\, X_{0jlp} \leq 1 \qquad \forall\, l\varepsilon K\varepsilon V \tag{15}$$

$$\Sigma_{j'\in N}\, X_{j'il(p-1)} - \Sigma_{j\in N}\, X_{ijlp} \leq 1 \qquad \forall\, i\varepsilon N,\ l\varepsilon K\varepsilon V,\, p\varepsilon P,\ p>1 \tag{16}$$

$$\Sigma_{i\in N} \Sigma_{j\in N} \Sigma_{p\in P}\, w_i X_{ijlp} \leq Qw_k \qquad \forall\, l\varepsilon K\varepsilon V \tag{17}$$

$$\Sigma_{i\in N} \Sigma_{j\in N} \Sigma_{p\in P}\, v_i X_{ijlp} \leq Qv_k \qquad \forall\, l\varepsilon K\varepsilon V \tag{18}$$

$$T_{il} - (ao_l + t_{oi})\, \Sigma_{p\in P} X_{0ilp} \geq 0 \qquad \forall\, i\varepsilon N,\, l\varepsilon K\varepsilon V \tag{19}$$

$$T_{jl} - T_{il} - (U_i + t_{ij})\, \Sigma_{p\in P} X_{ijlp} \geq 0 \qquad \forall\, i\varepsilon N,\, l\varepsilon K\varepsilon V \tag{20}$$

$$T_{il} - a_i\, \Sigma_{j\in N} \Sigma_{p\in P} X_{ijlp} \geq 0 \qquad \forall\, i\varepsilon N,\, l\varepsilon K\varepsilon V \tag{21}$$

$$T_{il} - b_i\, \Sigma_{j\in N} \Sigma_{p\in P} X_{ijlp} \leq 0 \qquad \forall\, i\varepsilon N,\, l\varepsilon K\varepsilon V \tag{22}$$

$$OT_l - T_{il} + e' \geq 0 \qquad \forall\, i\varepsilon N,\, l\varepsilon K\varepsilon V \tag{23}$$

$$OT_l - e + e' \leq 0 \qquad \forall\, l\varepsilon K\varepsilon V \tag{24}$$

$$X_{ijlp} \ \varepsilon \{0,1\} \qquad \forall\, i\varepsilon N,\ j\varepsilon N,\ l\varepsilon K\varepsilon V,\, p\varepsilon P \tag{25}$$

$$T_{il} \geq 0 \qquad \forall\, i\varepsilon N,\, l\varepsilon K\varepsilon V \tag{26}$$

$$OT_l \geq 0 \qquad \forall\, l\varepsilon K\varepsilon V \tag{27}$$

Fig. 21. The mathematical model for CVRPTWRO

Heuristics for CVRPTW

Two heuristics are developed, tested and implemented in to the routing system. The methods are outlined below. While following the steps of *RouteOrder* command (as explained in the graphical user interface subsection), a dialog box provides an option to choose between the heuristics to solve the VRPTW. Both the heuristics can be used to solve and the one that results in a lower cost can be used for vehicle assignments. The following subsections describe the two heuristics implemented in the routing system.

Nearest customer first heuristics

This heuristics as shown in Fig. 22 adds customers to a route starting with the RDC and choosing a customer based on the time window values and nearest distance to the current location, until either the capacity constraint or the time limit is violated. The heuristics starts again with a new route for unassigned customer locations. This process continues until there is no customer to be routed.

Farthest customer first heuristics

This heuristics is very similar to the one above. However, it differs from the previous one in selection of a candidate customer location for adding into a route. First, the customer orders are sorted according to the time-window values. Within this sort, a sub-sort based on distance from the RDC is performed. The delivery route is then designed in such a way that the vehicle travels to a farthest delivery location and works it way back home while delivering the remaining customer locations in its route. This logic is very much in practice especially in metropolitan area such as Dallas, TX due to peak hour traffic. Hence, this technique is particularly suitable for this RDC. The heuristics is detailed in Fig. 23.

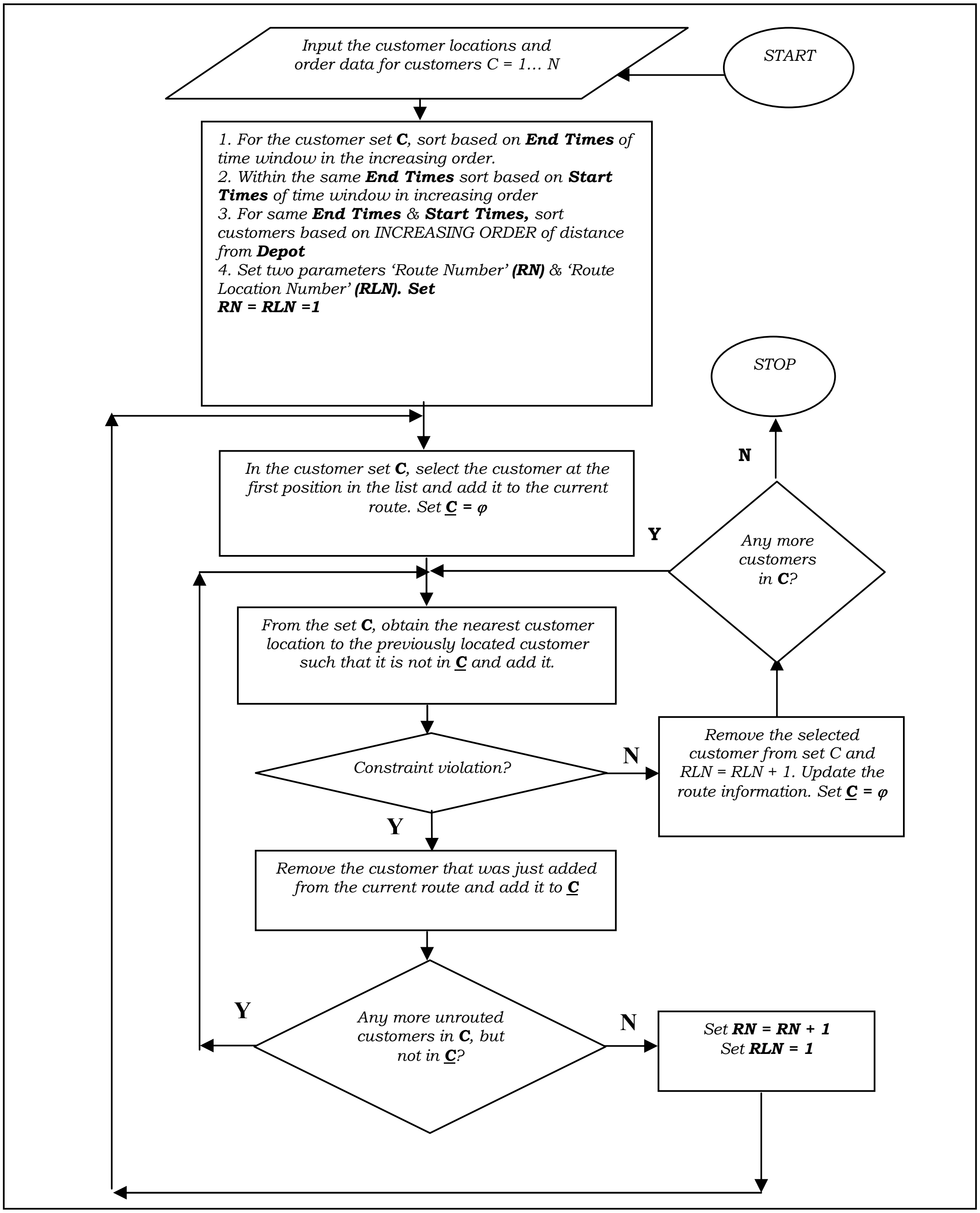


Fig. 22. Nearest customer heuristics

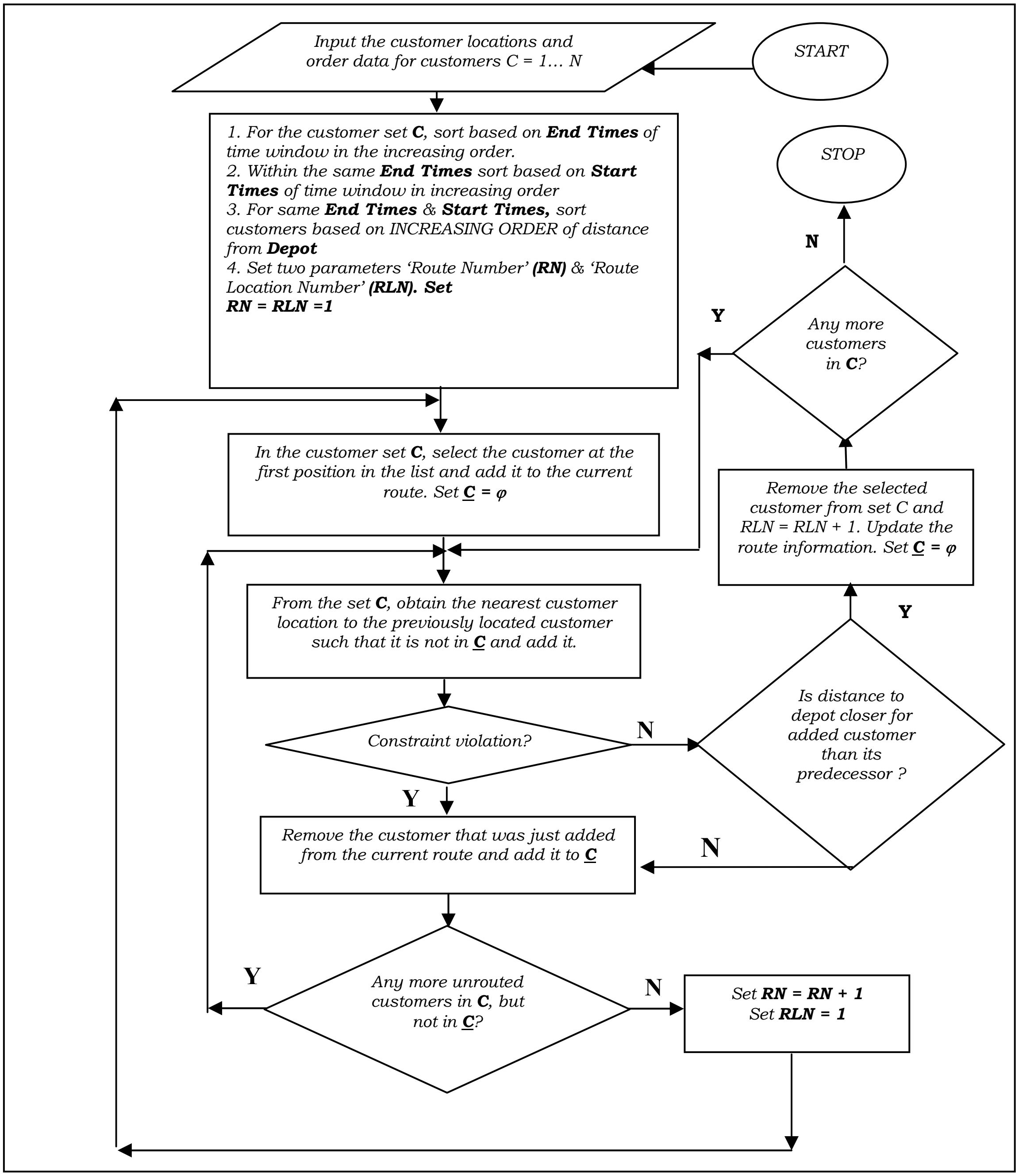


Fig. 23. Farthest customer heuristics

Re-optimization/What-if Analysis Module

The following subsection details the implementation of re-optimization module which aids the dispatcher in rush order processing. Supposing that the dispatcher has already routed few customer orders and instructed the staging dock to load orders, and later receives few rush-orders to be delivered along with the routed orders, then this module should be used to find a feasible and starting with partial routes. Fig. 24 shows the algorithm flow for re-optimization upon rush-order arrivals.

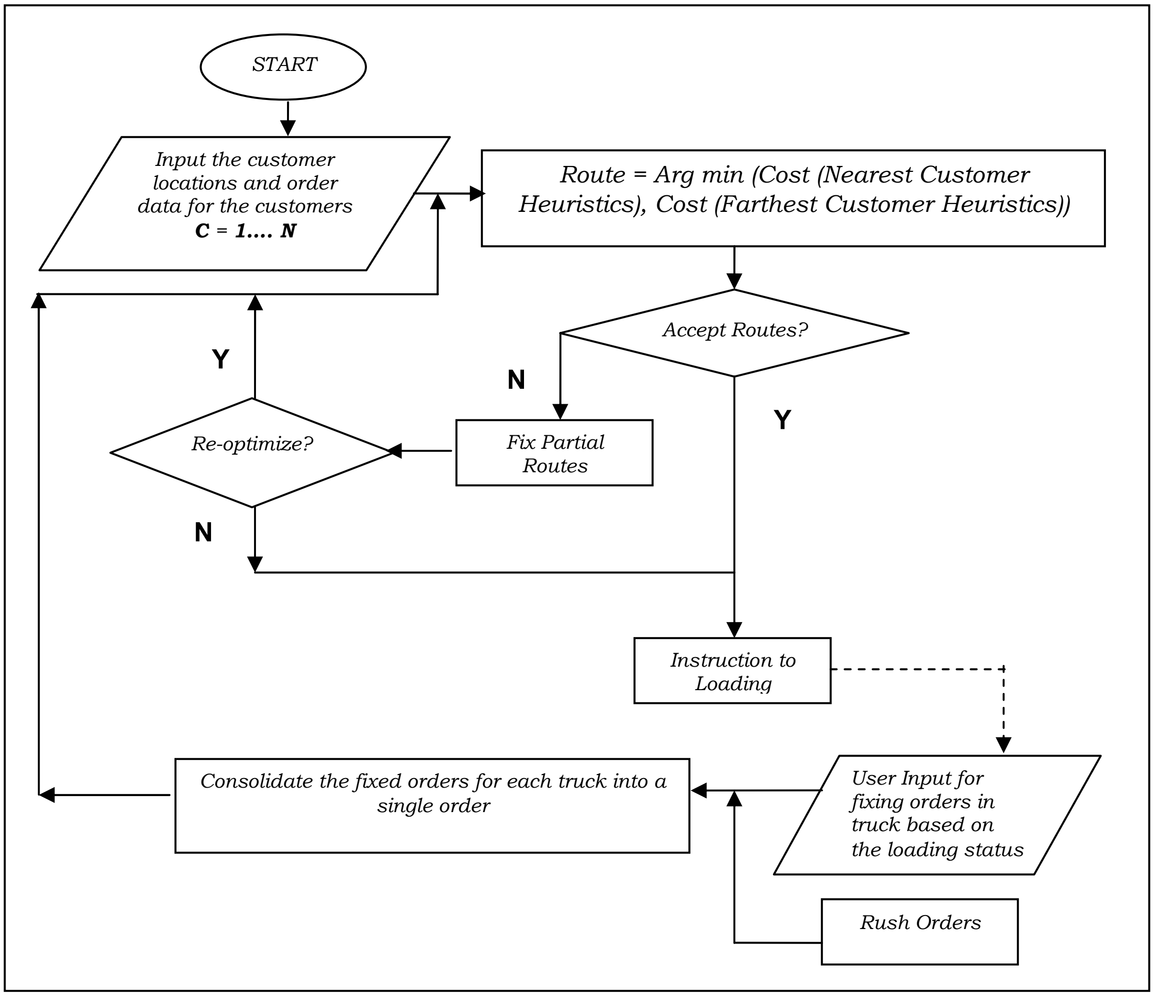
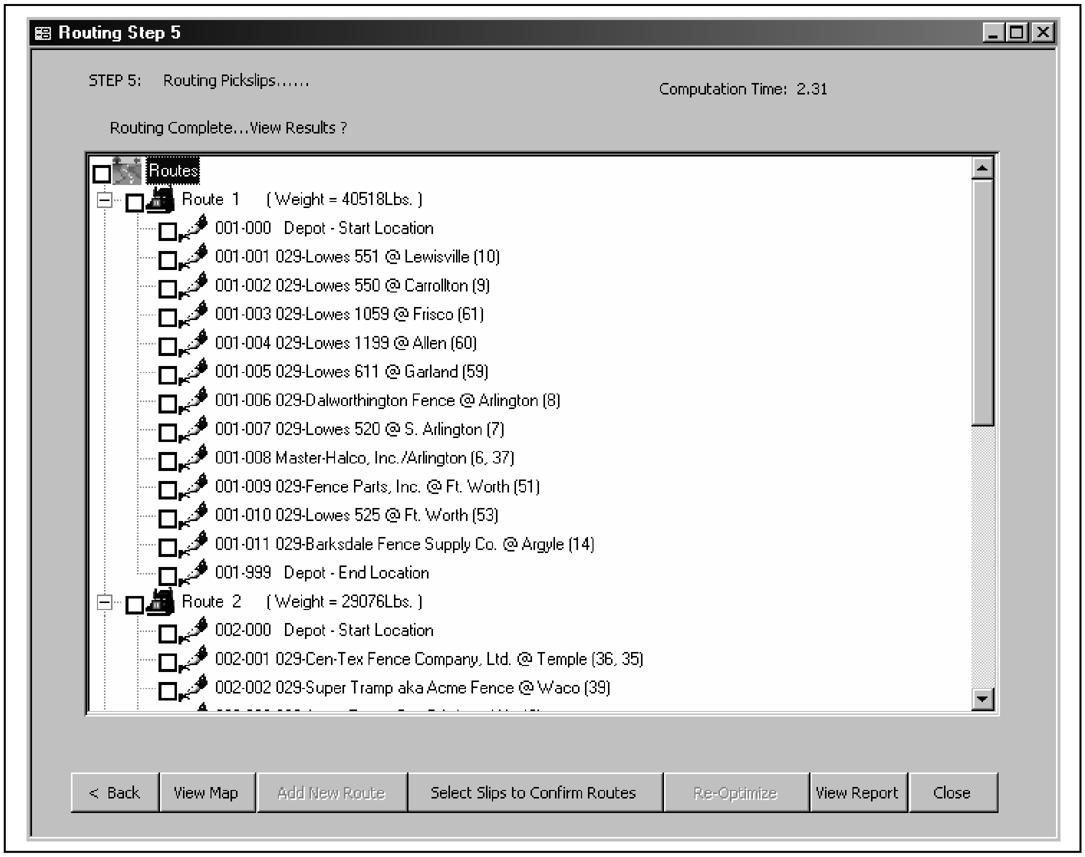


Fig. 24. Rush-order processing algorithm

This module accomplishes the rush order processing and performing what-if analysis in two steps. First, the customer orders that should not be removed from its route are *confirmed* by selecting the checkbox corresponding to the customer orders and selecting *Select Slips to Confirm Routes* command button on the form shown in Fig 25. The second step allows addition of new routes through *Add New Route* command to allow for the manipulation of on the results, for example, say to remove a customer order from a route and re-optimize based on new additions. Further, the form provides a drag and drop feature to drag a customer order from on route and drop into another route. During this process, the routing system checks for capacity, time-window violations on the route.



STEP 5: Routing Pickslips......

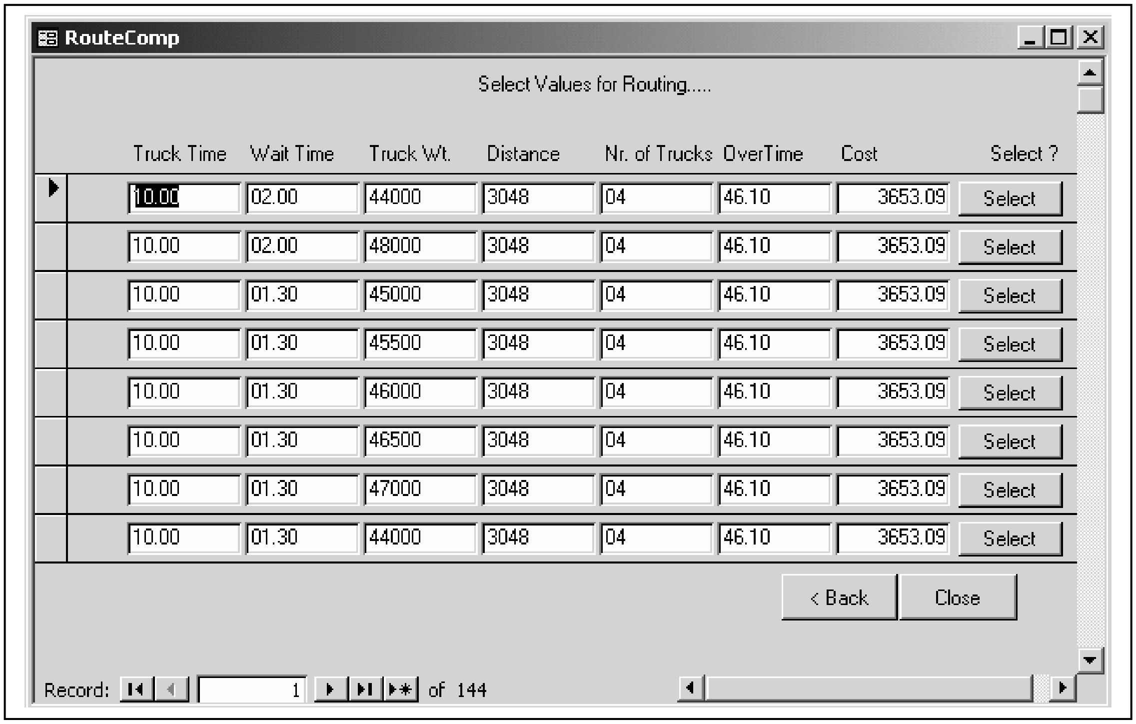
Computation Time: 2.31

Routing Complete...View Results ?

☐ Routes
- ☐ Route 1 (Weight = 40518Lbs.)
  - ☐ 001-000 Depot - Start Location
  - ☐ 001-001 029-Lowes 551 @ Lewisville (10)
  - ☐ 001-002 029-Lowes 550 @ Carrollton (9)
  - ☐ 001-003 029-Lowes 1059 @ Frisco (61)
  - ☐ 001-004 029-Lowes 1199 @ Allen (60)
  - ☐ 001-005 029-Lowes 611 @ Garland (59)
  - ☐ 001-006 029-Dalworthington Fence @ Arlington (8)
  - ☐ 001-007 029-Lowes 520 @ S. Arlington (7)
  - ☐ 001-008 Master-Halco, Inc./Arlington (6, 37)
  - ☐ 001-009 029-Fence Parts, Inc. @ Ft. Worth (51)
  - ☐ 001-010 029-Lowes 525 @ Ft. Worth (53)
  - ☐ 001-011 029-Barksdale Fence Supply Co. @ Argyle (14)
  - ☐ 001-999 Depot - End Location
- ☐ Route 2 (Weight = 29076Lbs.)
  - ☐ 002-000 Depot - Start Location
  - ☐ 002-001 029-Cen-Tex Fence Company, Ltd. @ Temple (36, 35)
  - ☐ 002-002 029-Super Tramp aka Acme Fence @ Waco (39)

Fig. 25. Route generation and re-optimization form

The what-if analysis tool is used to study the change in cost by varying parameters such as the truck time available for routing, the weight capacity, and the wait time (the time when a truck reaches the delivery site and waits for the customer due to time window constraints). This is achieved by executing the algorithm by varying the parameters of speed, truck time and capacity in certain step size and computing the total routing costs, overtime and number of trucks. The results of the iterations are then presented to the despatcher, as shown in Fig. 26, from which he can select a particular routing parameter based on either on the minimum cost, minimum number of trucks, less overtime (to reduce strain on drivers) or a trade-off among the above parameters.



Truck Time	Wait Time	Truck Wt.	Distance	Nr. of Trucks	OverTime	Cost	Select ?
10.00	02.00	44000	3048	04	46.10	3653.09	Select
10.00	02.00	48000	3048	04	46.10	3653.09	Select
10.00	01.30	45000	3048	04	46.10	3653.09	Select
10.00	01.30	45500	3048	04	46.10	3653.09	Select
10.00	01.30	46000	3048	04	46.10	3653.09	Select
10.00	01.30	46500	3048	04	46.10	3653.09	Select
10.00	01.30	47000	3048	04	46.10	3653.09	Select
10.00	01.30	44000	3048	04	46.10	3653.09	Select

Fig. 26. The results of variation in parameters with variation in cost

Thus the what-if analysis tool provides an overview of impact of variation of algorithm parameters, as defined in the parameter domain, on the costs, number of trucks, and overtime.

COMPUTATIONAL RESULTS

This section details the computational results of the AMPL/CPLEX code corresponding to the mathematical formulation shown in Fig. 21 and the two heuristics discussed in the previous section. Table 1 shows the results of test runs for smaller instances of data with number of customers ranging from three to eight. The objective values of the heuristics are compared with the optimal objective values. It can be seen that the runtime for the mixed integer program (MIP) increases exponentially with an increase in number of customers. However, both the heuristics require only a modest amount of time to find a feasible solution. The relative performance of the heuristics with respect to the optimal objective value of the MIP is also tabulated for comparison.

Table 1
Results of the MIP and the heuristics for smaller data sets

# of Customers	Run Time (secs)			Objective Value (Total Cost in $)			Relative Performance = Heuristics cost/Optimal value	
	MIP[a]	H1[a]	H2[a]	MIP[a]	H1[a]	H2[a]	H1[a]	H2[a]
3	0.31	0.12	0.12	129.73	135.92	133.68	1.048	1.030
3	0.28	0.12	0.12	241.20	241.20	241.20	1.000	1.000
4	6.30	0.12	0.12	273.56	280.72	280.32	1.026	1.025
4	0.55	0.12	0.12	277.96	286.72	280.32	1.032	1.008
5	5.80	0.12	0.12	380.99	516.37	492.25	1.355	1.292
5	1.60	0.12	0.12	660.89	665.24	680.43	1.007	1.030
6	51.00	0.15	0.18	754.69	763.35	763.35	1.011	1.011
6	65.00	0.20	0.21	683.71	825.68	743.62	1.208	1.088
7	2594.00	0.20	0.20	611.05	727.20	751.36	1.190	1.230
7	2416.00	0.20	0.25	484.45	681.76	632.81	1.407	1.306
8	11310.00	0.34	0.31	507.75	756.32	817.04	1.490	1.609
8	13582.00	0.34	0.32	605.82	830.96	814.36	1.372	1.344

[a] MIP: Mixed Integer Program Formulation; H1: Nearest Customer Heuristic; H2: Farthest Customer Heuristic

Instances of rush-orders were generated and added to the existing routes to study the effect of repositioning/reloading costs on the total cost. Table 2 shows the variation in total cost with variations in repositioning/reloading costs. Three different values of reloading costs were chosen for comparison. Column RC1 of Table 2 shows the objective values obtained by allowing reloading at no cost, whereas column RC2 shows objective values obtained by fixing the partially loaded trucks by assigning an infinite penalty cost for reloading. Column RC3 uses a reloading cost of $7.50 per order, which is derived based on the average unloading/reloading time of 30 minutes per order and an overtime cost of $15.00 per hour. The relative performance of the nearest customer first heuristics with respect to the optimal objective values with no reloading costs and infinite reloading cost are listed in columns RP1 and RP2 respectively. Similarly, the relative performance of the farthest customer first heuristics is tabulated in columns RP3 and RP4.

From Table 2, it can be seen that the relative performance of the heuristics with respect to the optimal objective values are consistent with the values in Table 1. Although there is a slight variation in the relative performance of the heuristics with the variation in the reloading costs, the average relative performance does not exhibit much variation. However, these results exhibit interesting characteristics leading to an argument, if rush order processing is necessary in the first place as compared to the policy where all the sales orders are accumulated till the end of the day and considered for routing as a single batch. The objective values corresponding to the case with no reloading costs dominate the total costs for all other cases, hence supporting the policy where there is no rush order processing. Further, this policy with no rush-order processing, will naturally eliminate the reloading nervousness in the system. However, results for larger instances of the company data needs to be benchmarked against the optimal solution to establish the superiority of this policy against the current practice that considers rush-orders.

Table 2
Effect of reloading costs on total cost due to rush-orders

Nr. of Orders	Nr. of Rush-orders	RC1[b] ($)					RC2[b] ($)					RC3[b] ($)
		MIP[a]	H1[a]	H2[a]	RP1[c]	RP3[c]	MIP[a]	H1[a]	H2[a]	RP2[c]	RP4[c]	MIP[a]
3	1	277.96	286.72	280.32	1.03	1.01	306.52	322.48	331.63	1.05	1.08	293.47
3	2	380.99	516.37	492.25	1.36	1.29	476.87	553.26	592.71	1.16	1.24	427.22
3	3	362.57	393.46	382.53	1.09	1.06	692.13	754.19	706.28	1.09	1.02	418.59
3	4	432.82	547.89	558.49	1.27	1.29	956.28	1103.44	1254.20	1.15	1.31	694.84
4	1	660.89	665.24	680.43	1.01	1.03	660.89	665.24	680.43	1.01	1.03	660.89
4	2	683.71	825.68	693.62	1.21	1.01	813.24	898.33	853.68	1.10	1.05	702.54
4	3	283.56	396.57	314.11	1.40	1.11	754.69	873.46	942.89	1.16	1.25	343.23
5	1	763.21	882.43	865.69	1.16	1.13	1058.61	1146.82	1254.40	1.08	1.18	874.54
5	2	623.83	710.66	763.84	1.14	1.22	989.24	1048.47	1023.30	1.06	1.03	688.32

[a] MIP: Mixed Integer Program Formulation; H1: Nearest Customer Heuristic; H2: Farthest Customer Heuristic

[b] RC1: reloading cost is $0; RC2: reloading cost is $∞ ; RC3: reloading cost is $ 7.50.

[c] RP1: (RC1 of H1)/(RC1 of MIP); RP2: (RC2 of H1)/(RC2 of MIP); RP3: (RC1 of H2)/(RC1 of MIP); RP2: (RC2 of H2)/(RC2 of MIP)

Average relative performance of heuristics: RP1:1.18; RP2: 1.09; RP3: 1.12; RP4: 1.14.

The dispatcher's routing based on rules-of-thumb is considered the benchmark for larger data instances. Table 3 shows the actual data as recorded from the company for two days. A sensitivity analysis along with the parameters used to solve the heuristics and the resulting total cost, over-time, total distance traveled and computation time required to solve are tabulated in Tables 4-5. Sample data containing the order information, customer inter-distance data are given Table 7 (Appendix A) and Table 8 (Appendix B) respectively. The runtime output of the AMPL/CPLEX during the branch and bound tree search is included in Table 9 (Appendix C). Also, the output as printed by the routing system for the company data is provided in Table 10 (Appendix D).

Table 3
Results from manual routing of company data

Days	Nr. of Customers	Avg. unload time (min)	Number of Trucks	Total Overtime (hr)	Total Cost
1	30	45	6	31.5	$2348
2	26	30	5	12.5	$1715

It can be seen from Table 4 that the farthest customer heuristic outperforms the nearest customer heuristic by yielding a percentage cost savings of 31.43 % on the first day and about 37.98% on the second day (results shown in Table 5) as compared to the benchmark solutions. The farthest customer first heuristic is found to perform better than the other when the customers are located at a larger distance from the RDC and there are very few customers located near the RDC. The nearest customer heuristic performs better when the converse is true.

Table 4

Sensitivity analysis I (total cost vs variations in vehicle capacity, maximum truck time, and average truck speed for first day data)

S. Nr.	Truck Parameters			Farthest Customer Heuristic (H2)					Nearest Customer Heuristic (H1)				
	Weight (lbs)	Time (hr)	Speed (mph)	# of Routes	Overtime (min)	Total Cost ($)	% Cost Savings[a]	Computation Time (s)	# of Routes	Overtime (min)	Total Cost ($)	% Cost Savings[a]	Computation Time (s)
1	44000	12	40	7	8.00	1890	19.51	3.47	7	9.25	1984	15.50	3.47
2	44000	12	45	7	6.10	1835	21.85	3.47	7	6.50	1928	17.89	3.47
3	44000	14	40	7	8.30	1889	19.55	2.31	7	9.25	1986	15.42	2.31
4	44000	14	45	7	6.45	1849	21.25	2.31	7	6.50	1928	17.89	3.47
5	44000	16	40	7	8.30	1889	19.55	2.31	7	9.25	1986	15.42	3.47
6	44000	16	45	7	6.45	1849	21.25	2.31	7	6.50	1920	18.23	3.47
7	48000	12	40	6	7.40	1673	28.75	2.31	6	5.45	1907	18.78	2.31
8	48000	12	45	6	4.50	1610	31.43	2.31	6	2.50	1840	21.64	3.47
9	48000	14	40	6	7.40	1673	28.75	2.31	6	9.20	2074	11.67	2.31
10	48000	14	45	6	11.00	1934	17.63	3.47	6	6.10	2018	14.05	3.47
11	48000	16	40	6	7.40	1673	28.75	3.47	6	9.20	2074	11.67	3.47
12	48000	16	45	6	11.00	1934	17.63	2.31	6	6.10	2018	14.05	2.31

[a] Refer to day 1 data from Table 3 for comparison. Average % Cost Savings 1. H2: 22.99% 2. H1: 16.01%

Table 5

Sensitivity analysis II (total cost vs variations in vehicle capacity, maximum truck time, and average truck speed for second day data)

S. Nr.	Truck Parameters			Farthest Customer Heuristic (H2)					Nearest Customer Heuristic (H1)				
	Weight (lbs)	Time (hr)	Speed (mph)	# of Routes	Overtime (min)	Total Cost ($)	% Cost Savings[a]	Computation Time (s)	# of Routes	Overtime (min)	Total Cost ($)	% Cost Savings[a]	Computation Time (s)
1	44000	12	40	4	6.25	1304	23.97	2.31	4	9.55	1409	17.84	1.15
2	44000	12	45	4	6.10	1322	22.92	2.31	4	7.30	1356	20.93	2.31
3	44000	14	40	4	8.30	1374	19.88	1.15	4	9.55	1409	17.84	2.31
4	44000	14	45	4	6.10	1322	22.92	2.31	4	7.30	1356	20.93	1.15
5	44000	16	40	4	8.30	1374	19.88	1.15	4	9.55	1409	17.84	1.15
6	44000	16	45	4	6.10	1322	22.92	2.31	4	7.30	1356	20.93	2.31
7	48000	12	40	4	8.55	1356	20.93	1.15	4	9.55	1409	17.84	1.15
8	48000	12	45	3	9.10	1203	29.85	2.31	3	7.10	1098	35.98	1.15
9	48000	14	40	3	11.40	1260	26.53	1.15	3	9.30	1149	33.00	2.31
10	48000	14	45	3	9.10	1203	29.85	1.15	3	7.10	1098	35.98	2.31
11	48000	16	40	3	11.40	1260	26.53	2.31	3	9.30	1149	33.00	1.15
12	48000	16	45	3	9.10	1203	29.85	2.31	3	7.10	1098	35.98	1.15

[a] Refer to day 2 data from Table 3 for comparison. Average % Cost Savings 1. Farthest Customer Heuristic: 24.66% 2. Nearest Customer Heuristic: 25.67%

However naturally, both the heuristics performed better than the manual routing on both days and for the range of the parameters chosen, there is at least an 11.67% cost savings on day one and about 17.84% cost savings on day two as compared with the manual routing solution. Table 6 shows the what-if analysis results for variation of truck weight capacity, total truck time and wait time (i.e. the time the truck can be allowed to wait at a customer delivery location if it reaches before the open time of the customer as specified in the time-window). For the analysis, the total truck time is varied from 8 hrs to 16 hrs in steps of an hour; the truck wait time is varied from 0.25 hrs (15 min) to 0.5 hrs (30 min) in a step of 0.25 hrs; and the weight capacity of truck is varied between 48000 lbs and 44000 lbs in steps of 2000lbs.

Table 6
What-if analysis results (for first day data sorted for increasing total cost)

Total truck hours (hrs)	Wait time (hr)	Weight capacity (lbs)	Total distance of routes (mi)	Number of routes	Total overtime (hrs)	Total cost ($)
15.00	00.30	48000	1112	06	07.40	1672.60
14.00	00.30	48000	1112	06	07.40	1672.60
13.00	00.30	48000	1112	06	07.40	1672.60
12.00	00.30	48000	1112	06	07.40	1672.60
16.00	00.30	48000	1112	06	07.40	1672.60
16.00	00.30	46000	1107	06	09.15	1703.83
15.00	00.30	46000	1107	06	09.15	1703.83
14.00	00.30	46000	1107	06	09.15	1703.83
13.00	00.30	46000	1107	06	09.15	1703.83
12.00	00.30	46000	1107	06	09.15	1703.83
09.00	00.30	48000	1239	07	01.50	1749.42
16.00	00.15	44000	1145	06	10.10	1751.56
14.00	00.15	44000	1145	06	10.10	1751.56
15.00	00.15	44000	1145	06	10.10	1751.56
11.00	00.30	44000	1168	07	04.50	1767.50
11.00	00.15	46000	1181	07	05.55	1801.96
11.00	00.15	44000	1181	07	05.55	1801.96
16.00	00.15	48000	1140	06	12.40	1803.19
16.00	00.15	46000	1140	06	12.40	1803.19
15.00	00.15	48000	1140	06	12.40	1803.19
14.00	00.15	48000	1140	06	12.40	1803.19
14.00	00.15	46000	1140	06	12.40	1803.19
15.00	00.15	46000	1140	06	12.40	1803.19

Table 6 Continued

Total truck hours (hrs)	Wait time (hr)	Weight capacity (lbs)	Total distance of routes (mi)	Number of routes	Total overtime (hrs)	Total cost ($)
11.00	00.30	48000	1177	07	07.00	1822.23
10.00	00.30	44000	1267	07	04.15	1824.00
13.00	00.15	44000	1158	07	08.05	1834.57
12.00	00.15	44000	1158	07	08.05	1834.57
11.00	00.30	46000	1225	07	06.05	1835.81
11.00	00.15	48000	1277	07	05.55	1869.23
10.00	00.30	48000	1292	07	06.05	1882.80
12.00	00.15	48000	1153	07	10.35	1886.21
12.00	00.15	46000	1153	07	10.35	1886.21
13.00	00.15	48000	1153	07	10.35	1886.21
13.00	00.15	46000	1153	07	10.35	1886.21
13.00	00.30	44000	1223	07	08.30	1888.57
14.00	00.30	44000	1223	07	08.30	1888.57
15.00	00.30	44000	1223	07	08.30	1888.57
16.00	00.30	44000	1223	07	08.30	1888.57
12.00	00.30	44000	1242	07	08.00	1890.14
10.00	00.15	46000	1296	07	06.40	1899.08
10.00	00.15	44000	1296	07	06.40	1899.08
10.00	00.15	48000	1304	07	06.40	1904.33
09.00	00.30	44000	1291	08	02.40	1925.63
09.00	00.15	46000	1335	08	02.20	1948.74
09.00	00.15	44000	1335	08	02.20	1948.74
09.00	00.30	46000	1340	08	02.45	1960.36
09.00	00.15	48000	1435	08	02.00	2010.29
10.00	00.30	46000	1415	08	04.05	2043.49
08.00	00.30	48000	1421	09	00.00	2075.19
08.00	00.30	46000	1453	09	00.00	2097.59
08.00	00.30	44000	1491	09	00.00	2123.76
08.00	00.15	44000	1518	09	00.00	2142.67
08.00	00.15	48000	1518	09	00.00	2142.67
08.00	00.15	46000	1518	09	00.00	2142.67

The above result aids the dispatcher in selecting a solution for vehicle assignment based either on minimum total cost, or lesser total overtime (less stressful for the drivers), or a lesser number of routes. In the following section, conclusions and recommendations for future research are presented.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Conclusions

This research involves the design and development of a routing system to solve capacitated vehicle routing problem with time-windows and reloading under rush-order constraints. The mathematical formulation using mixed integer programming is attempted to solve the problem using CPLEX solver and AMPL interface. Test runs confirmed that the problem is difficult to be solved and requires an exponential time as number of customers, width of the time-windows, and number of vehicles. Hence two heuristics, one based on nearest customer clustering and the other based on farthest customer clustering is developed, tested and implemented in the system. Test runs were made for several days and results for two days are reported. The heuristics consistently performed better than manual routing by dispatcher. Rush order processing is carried out by fixing the loaded customer orders to a specific route and allowing the remaining orders to be routed by the algorithm iteratively. Support functions such as inter-distance learning/maintenance tool, GIS interface are developed and implemented in the system. The following subsection outlines recommendations for future research and improvements for solving the VRPTWRO and its extensions.

Recommendations for Future Research

The computational results of the routing algorithm used to solve the VRPTWRO provide a superior and cost effective solution for the company. However, these heuristics are not guaranteed to obtain optimal or near optimal solutions. Recent advances in solution methods such as branch-and-price, branch, price, and cut, meta-heuristics, and evolutionary algorithms can be used to solve the VRPTWRO for solutions better than those obtained with the heuristics.

Further, the problem can be generalized to include pick-ups from customer location (reverse logistics). Also, constraints such as material handling equipment requirements at the delivery sites can be added into the model to generalize the current problem. These are the few recommendations for future research that can improve and solve real world problems of the RDC of the company.

REFERENCES

Beasley, J., 1983. Route first – cluster second methods for vehicle routing. Omega 11, 403-408.

Bianco, L., Ricciardelli, S., Rinaldi, G., Sassano, A., 1988. Scheduling tasks with sequence-dependent processing times. Naval Research Logistics Quarterly 35, 177-184.

Blanton, J. L., Wainwright, R. L., 1993. Multiple vehicle routing with time and capacity constraints using genetic algorithms. Proceedings of the 5th International Conference on Genetic Algorithms, San Mateo, CA. 452-459.

Bodin, L., Golden, B., 1981. Classification in vehicle routing and scheduling. Networks 11, 97-108.

Bramel, J., Simchi-Levi, D., 1995. A location based heuristic for general routing problems. Operations Research 43, 649-660.

Bramel, J., Simchi-Levi, D., 1997. The Logic of Logistics. Springer-Verlag, New York.

Carlton, W. B., 1995. A tabu search approach to the general vehicle routing problem. Ph.D. Thesis, The University of Texas at Austin, TX.

Chiang, W. C., Russell, R. A., 1996. Simulated annealing metaheuristics for the vehicle routing problem with time-windows. Annals of Operational Research 63, 3-27.

Christofides, N., Mingozzi, A., Toth, P., 1978. The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C., (Eds.), Combinatorial Optimization. John Wiley & Sons Ltd., New York. 318-338.

Clark, G., Wright, J. W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568-581.

Cordeau, J. F., Desaulniers, G., Desrosiers, J., Solomon, M. M., Soumis F., 2002. The VRP with time windows. In: Toth, P., Vigo, Daniel (Eds.), The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications 9, 157-193.

Desrochers, M., Desrosiers, J., Solomon, M.M., 1992. A new optimization algorithm for the vehicle routing problem with time-windows. Operations Research 40, 342-354.

Fisher, M. L., 1994. Optimal solution of the vehicle routing problems using minimum k-tress. Operations Research 42, 626-642.

Fisher, M. L., Jaikumar, R., 1981. A generalized assignment heuristic for vehicle routing. Networks 11, 109-124.

Gambardella L.M, Taillard, E., Agazzi, G., 1999. MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. In: Corne, D., Dorigo, M., Glover, F., (Eds.), New Ideas in Optimization, McGraw-Hill, London, UK. 63-76.

Gillett, B. E., Miller, L. R., 1974. A heuristic algorithm for the vehicle dispatch problem. Operations Research 22, 340-349.

Kallehauge, B., 2001. Lagrangian duality and non-differentiable optimization applied on routing with time-windows. M.Sc. Thesis IMM-EKS-2000-13, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Koskosidis, Y. A., Powell, W. B., Solomon, M. M., 1992. An optimization based heuristic for vehicle routing and scheduling with soft time-window constraints. Transportation Science 26, 69-85.

PCMILER STREETS 3.0. ALK Technologies. http://www.pcmiler.com or http://www.alk.com/products/business/trucking/pcmiler_streets/index.asp, Accessed June 30, 2003.

Pinedo, M., 2002. Scheduling: Theory, Algorithms, and Systems. Prentice Hall, Upper Saddle River, New Jersey.

Potvin, J. Y., Kervahut, T., Garcia, B. L., Rousseau, J. M., 1996. The vehicle routing problem with time-windows – Part I: tabu search. INFORMS Journal on Computing 8, 158-164.

Potvin, J. Y., Bengio, S., 1996. The vehicle routing problem with time-windows – Part II: genetic search. INFORMS Journal on Computing 8, 165-172.

Savelsberg, M. W.P., 1985. Local search in routing problems with time windows. Annals of Operations Research 4, 285-305.

Solomon, M. M., Desrosiers, J., 1988. Time window constrained routing and scheduling problems. Transportation Science 22, 1-13.

Tang, C. S., 1990. Scheduling batches on parallel machines with major and minor setups. European Journal of Operational Research 46, 28-37.

Wittrock, R. J., 1990. Scheduling algorithms for flexible flow lines. IBM Journal of Research and Development 29, 401-412.

APPENDIX A

Table 7
Customer order data

Pick-slip	CustomerNr	DeliveryDate	TotWt	IncludeSlip	DepotDist	LocNo	Open	Close	Name	TotVol
1	56042	10/21/2002	12030	Yes	27.9	24	11	13	029-Diamond B Fence Company @ Arlington	1971.58
2	56042	10/21/2002	24974	Yes	27.9	24	11	13	029-Diamond B Fence Company @ Arlington	3767.83
3	56042	10/21/2002	328	Yes	27.9	24	11	13	029-Diamond B Fence Company @ Arlington	2866.06
4	56042	10/21/2002	8474	Yes	27.9	24	11	13	029-Diamond B Fence Company @ Arlington	1362.47
6	12001	10/21/2002	1408	Yes	27.3	1	8	17	Master-Halco, Inc./Arlington	229.91
7	141680	10/21/2002	3130	Yes	32.9	2	10	13	029-Lowes 520 @ S. Arlington	383.03
8	55693	10/21/2002	7551	Yes	28.2	13	8	17	029-Dalworthington Fence @ Arlington	4020.07
9	142576	10/21/2002	3790	Yes	26.4	4	7	10	029-Lowes 550 @ Carrollton	3259.02
10	142388	10/21/2002	10991	Yes	30.6	3	7	10	029-Lowes 551 @ Lewisville	2532.74
11	55911	10/21/2002	12296	Yes	36.5	21	8	17	029-All Texas Fence @ Lake Dallas	1854.75
12	55911	10/21/2002	8271	Yes	36.5	21	8	17	029-All Texas Fence @ Lake Dallas	1395.31
13	55911	10/21/2002	863	Yes	36.5	21	8	17	029-All Texas Fence @ Lake Dallas	3992.82
14	55728	10/21/2002	663	Yes	51	14	8	17	029-Barksdale Fence Supply Co. @ Argyle	2921.30
15	55798	10/21/2002	470	Yes	48.4	18	8	17	029-Hurricane Fence Company @ Denton	1123.46

Table 7 Continued

Pick-slip	CustomerNr	DeliveryDate	TotWt	IncludeSlip	DepotDist	LocNo	Open	Close	Name	TotVol
16	55798	10/21/2002	11037	Yes	48.4	18	8	17	029-Hurricane Fence Company @ Denton	1129.04
17	55798	10/21/2002	6008	Yes	48.4	18	14	17	029-Hurricane Fence Company @ Denton	2146.15
18	55798	10/21/2002	139	Yes	48.4	18	8	17	029-Hurricane Fence Company @ Denton	1398.92
20	55804	10/21/2002	7127	Yes	107.7	19	8	17	029-Emerson Fence Co. @ Paris	2860.98
21	265078	10/21/2002	10266	Yes	74.1	7	8	17	029-Nathaniel Fence @ Sulphur Springs	2076.24
22	64308	10/21/2002	3386	Yes	74	29	8	17	029-Foxworth 42 @ Sulpher Springs	1377.32
24	56156	10/21/2002	20151	Yes	73.3	26	10	17	029-East Texas Landscaping @ Sulphur Spr	4751.20
25	56156	10/21/2002	1010	Yes	73.3	26	10	17	029-East Texas Landscaping @ Sulphur Spr	3128.15
26	64301	10/21/2002	415	Yes	112.1	28	8	17	029-FOXWORTH 25 @ PARIS	280.52
27	55817	10/21/2002	451	Yes	104.3	20	8	17	029-Metro Gate & Manufacturing @ Paris	772.33
28	55817	10/21/2002	174	Yes	104.3	20	8	17	029-Metro Gate & Manufacturing @ Paris	697.55
29	55817	10/21/2002	6	Yes	104.3	20	8	17	029-Metro Gate & Manufacturing @ Paris	623.54
30	55817	10/21/2002	338	Yes	104.3	20	8	17	029-Metro Gate & Manufacturing @ Paris	1804.04
31	55817	10/21/2002	4	Yes	104.3	20	8	17	029-Metro Gate & Manufacturing @ Paris	832.79
32	55992	10/21/2002	2	Yes	89	22	8	17	029-Mid-Tex Fence & Repair @ Elm Mott	705.87
33	55992	10/21/2002	33	Yes	89	22	8	17	029-Mid-Tex Fence & Repair @ Elm Mott	117.25

Table 7 Continued

Pick-slip	CustomerNr	DeliveryDate	TotWt	IncludeSlip	DepotDist	LocNo	Open	Close	Name	TotVol
34	55992	10/21/2002	1216	Yes	89	22	8	17	029-Mid-Tex Fence & Repair @ Elm Mott	4177.03
35	55791	10/21/2002	9475	Yes	134.7	17	8	17	029-Cen-Tex Fence Company, Ltd. @ Temple	899.00
36	55791	10/21/2002	7128	Yes	134.7	17	8	17	029-Cen-Tex Fence Company, Ltd. @ Temple	12663.00
37	12001	10/21/2002	4257	Yes	27.3	1	8	17	Master-Halco, Inc./Arlington	2711.00
38	55993	10/21/2002	5145	Yes	35.6	23	8	17	029-PWS Renovations @ Waxahachie	1289.00
39	268984	10/21/2002	3037	Yes	100.8	9	8	17	029-Super Tramp aka Acme Fence @ Waco	6155.00
40	56110	10/21/2002	1880	Yes	100.5	25	15	17	029-Arrow Fence Co. @ Waco	8790.00
41	56110	10/21/2002	440	Yes	100.5	25	15	17	029-Arrow Fence Co. @ Waco	4396.88
42	268351	10/21/2002	3134	Yes	111.6	8	8	17	029-Devco Fence and Supply @ Waco	1164.00
43	268351	10/21/2002	5	Yes	111.6	8	8	17	029-Devco Fence and Supply @ Waco	288.00
44	268351	10/21/2002	355	Yes	111.6	8	8	17	029-Devco Fence and Supply @ Waco	5040.00
45	55789	10/21/2002	720	Yes	93.2	16	8	17	029-Brem's Fencing & Repair, Inc. @ Waco	300.00
46	7200	10/21/2002	971	Yes	37.8	30	8	17	Master Halco, Inc./Ft. Worth	1582.88
47	7200	10/21/2002	192	Yes	37.8	30	8	17	Master Halco, Inc./Ft. Worth	3517.50
48	7200	10/21/2002	1935	Yes	37.8	30	8	17	Master Halco, Inc./Ft. Worth	3693.38
49	7200	10/21/2002	7681	Yes	37.8	30	8	17	Master Halco, Inc./Ft. Worth	2388.00

Table 7 Continued

Pick-slip	CustomerNr	DeliveryDate	TotWt	IncludeSlip	DepotDist	LocNo	Open	Close	Name	TotVol
51	55678	10/21/2002	28	Yes	41.4	12	8	17	029-Fence Parts, Inc. @ Ft. Worth	7035.00
53	185030	10/21/2002	1357	Yes	45.8	5	8	17	029-Lowes 525 @ Ft. Worth	8280.78
54	55741	10/21/2002	64	Yes	20.5	15	9	12	029-North Texas Crown Fence Co. @ Wylie	234.50
55	55741	10/21/2002	23	Yes	20.5	15	9	12	029-North Texas Crown Fence Co. @ Wylie	87.94
56	55741	10/21/2002	299	Yes	20.5	15	9	12	029-North Texas Crown Fence Co. @ Wylie	6070.00
57	55741	10/21/2002	45	Yes	20.5	15	9	12	029-North Texas Crown Fence Co. @ Wylie	87.94
58	55741	10/21/2002	45030	Yes	20.5	15	9	12	029-North Texas Crown Fence Co. @ Wylie	25600.00
59	215725	10/21/2002	4186	Yes	11.9	6	8	17	029-Lowes 611 @ Garland	48000.00
60	270107	10/21/2002	1694	Yes	29.4	11	8	17	029-Lowes 1199 @ Allen	2888.00
61	269277	10/21/2002	1463	Yes	28.8	10	8	17	029-Lowes 1059 @ Frisco	6000.00
62	64285	10/21/2002	1249	Yes	66.1	27	9	15	029-Allied Fence of Sherman @ Sherman	1006.00
63	64285	10/21/2002	264	Yes	66.1	27	9	15	029-Allied Fence of Sherman @ Sherman	1222.63

APPENDIX B

Table 8
Customer inter-distance data

Location Nr	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	5	34	40	21	49	112	109	97	50	54	18	7	43	59	90	132	51	194	168	49	85	31	9	98	112	94	165	116	20
2	5	0	30	36	20	45	108	109	96	47	50	17	4	42	56	89	131	50	191	164	46	85	42	5	97	108	90	162	112	19
3	34	30	0	11	38	29	99	132	119	15	22	32	29	16	36	113	154	19	133	134	8	108	57	32	121	100	60	131	103	26
4	40	36	11	0	45	18	93	126	114	14	20	39	34	24	30	107	148	26	132	133	15	102	52	36	115	92	68	130	97	33
5	21	20	38	45	0	57	116	104	92	51	61	7	22	34	64	84	128	42	175	176	51	80	61	20	93	116	106	174	120	13
6	49	45	29	18	57	0	91	132	116	19	13	51	43	45	8	111	154	46	127	128	36	106	58	45	118	91	56	124	97	44
7	112	108	99	93	116	91	0	188	175	96	73	112	103	114	98	168	210	116	45	33	104	164	113	104	176	1	87	42	5	109
8	109	109	132	126	104	132	188	0	11	132	134	104	110	129	136	21	45	137	270	247	137	24	83	108	13	188	176	245	192	108
9	97	96	119	114	92	116	175	11	0	120	121	91	98	117	124	8	35	125	258	235	125	12	71	95	2	175	164	259	179	95
10	50	47	15	14	51	19	96	132	120	0	9	45	46	28	24	112	155	31	120	121	20	108	58	47	121	96	50	118	100	39
11	54	50	22	20	61	13	73	134	121	9	0	55	50	35	15	114	156	38	114	115	27	110	59	50	122	73	41	112	77	48
12	18	17	32	39	7	51	112	104	91	45	55	0	17	28	58	84	128	36	170	170	45	80	59	15	93	112	98	168	116	7
13	7	4	29	34	22	43	103	110	98	46	50	17	0	41	52	91	133	49	185	162	36	86	45	3	99	103	92	159	107	14
14	43	42	16	24	34	45	114	129	117	28	35	28	41	0	52	109	153	11	145	145	12	105	72	38	118	114	75	143	118	31
15	59	56	36	30	64	8	98	136	124	24	15	58	52	52	0	117	159	54	133	134	42	113	62	52	125	97	60	131	102	51
16	90	89	113	107	84	111	168	21	8	112	114	84	91	109	117	0	44	118	250	227	117	4	64	89	10	169	157	226	171	88
17	132	131	154	148	128	154	210	45	35	155	156	128	133	153	159	44	0	160	294	271	161	48	106	130	35	210	199	267	215	130
18	51	50	19	26	42	46	116	137	125	31	38	36	49	11	54	118	160	0	40	120	134	66	116	126	46	74	113	12	137	136
19	194	191	133	132	175	127	45	270	258	120	114	170	185	145	133	250	294	40	0	13	139	247	196	187	259	44	73	5	42	162
20	168	164	134	133	176	128	33	247	235	121	115	170	162	145	134	227	271	120	13	0	140	225	173	164	237	32	74	11	31	163
21	49	46	8	15	51	36	104	137	125	20	27	45	36	12	42	117	161	134	139	140	0	114	63	40	127	105	68	137	109	34
22	85	85	108	102	80	106	164	24	12	108	110	80	86	105	113	4	48	66	247	225	114	0	60	84	13	164	153	221	169	84
23	31	42	57	52	61	58	113	83	71	58	59	59	45	72	62	64	106	116	196	173	63	60	0	47	72	113	102	170	117	60
24	9	5	32	36	20	45	104	108	95	47	50	15	3	38	52	89	130	126	187	164	40	84	47	0	97	104	90	161	108	11
25	98	97	121	115	93	118	176	13	2	121	122	93	99	118	125	10	35	46	259	237	127	13	72	97	0	177	165	261	181	97
26	112	108	100	92	116	91	1	188	175	96	73	112	103	114	97	169	210	74	44	32	105	164	113	104	177	0	87	41	4	110
27	94	90	60	68	106	56	87	176	164	50	41	98	92	75	60	157	199	113	73	74	68	153	102	90	165	87	0	71	92	91
28	165	162	131	130	174	124	42	245	259	118	112	168	159	143	131	226	267	12	5	11	137	221	170	161	261	41	71	0	40	160
29	116	112	103	97	120	97	5	192	179	100	77	116	107	118	102	171	215	137	42	31	109	169	117	108	181	4	92	40	0	114
30	20	19	26	33	13	44	109	108	95	39	48	7	14	31	51	88	130	136	162	163	34	84	60	11	97	110	91	160	114	0

APPENDIX C

Table 9
AMPL/CPLEX Log File

```
ampl: model rp.mod;
ampl: data RpNew.dat;
ampl: option cplex_options 'timing=1';
ampl:  option cplex_options 'mipdisplay=2';
ampl: solve;
CPLEX 6.5.2: mipdisplay=2

Clique table members: 1105
Root relaxation solution time =    0.28 sec.

        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer     Best Node    ItCnt     Gap

      0     0      311.7333   249                   311.7333      588
                   311.7334   254                   Cuts:  6      787
    100   100      312.3895   242                   311.7334     1758
    200   200      313.2493   170                   311.7334     2283
    300   296      450.4942    71                   311.7337     6867
    400   380      398.3031    85                   311.8241    13136
    500   470      440.2301    69                   311.8564    17445
    600   564      446.8287    75                   311.8600    22846
    700   640      456.2745    39                   312.1320    27212
    800   714    infeasible                         312.2160    34877
    900   794      534.9050    94                   312.2544    42316
   1000   880      460.0677   121                   312.2717    49184
   1100   980      597.0205    28                   312.2717    51155
*  1129   992      674.2206     0      674.2206     312.3238    55177   53.68%
   1200  1041      643.9448     4      674.2206     312.4027    64686   53.66%
*  1200+ 1041      643.9967     0      643.9967     312.4027    64686   51.49%
   1700  1336      399.2157   101      643.9967     316.4338   178066   50.86%
   1800  1423      391.7966    66      643.9967     318.0205   183128   50.62%
   1900  1497    infeasible            643.9967     376.3341   193699   41.56%
   2000  1551      395.3808   126      643.9967     386.4646   197333   39.99%
   3000  2068      396.4575    97      643.9967     396.4575   318172   38.44%
   3100  2167      403.5408    93      643.9967     396.4840   319757   38.43%
   3700  2281    infeasible            643.9967     399.4370   402582   37.98%
*  4848  2787      632.1268     0      632.1268     404.1928   528601   36.06%
   5000  2833      409.6374    95      632.1268     404.9997   543029   35.93%
   6000  3084      444.2762   104      632.1268     410.7621   649923   35.02%
   7000  3332      422.0668    68      632.1268     414.7215   779064   34.39%
   8000  3698      442.0203    94      632.1268     420.4161   874741   33.49%
   8900  4078      448.9190    83      632.1268     424.1114   962511   32.91%
   9000  4043      524.8292    71      632.1268     424.7027   977185   32.81%
  10000  4382        cutoff            632.1268     427.9303  1097143   32.30%
  10900  4825      626.3773     8      632.1268     429.8457  1231295   32.00%
  11000  4812      549.4842    76      632.1268     430.6287  1252581   31.88%
  12300  5137        cutoff            632.1268     435.2823  1449581   31.14%
* 12364  4986      611.0516     0      611.0516     435.4852  1462511   28.73%
  14100  5534        cutoff            611.0516     439.9853  1665044   28.00%
  14200  5577      556.5978    44      611.0516     440.1562  1681056   27.97%
  14300  5536    infeasible            611.0516     440.6254  1700613   27.89%
  14400  5528      462.6140    94      611.0516     440.8623  1710563   27.85%
  14500  5546      506.7532    21      611.0516     441.0933  1722317   27.81%
  14600  5546      558.6953    49      611.0516     441.4135  1736900   27.76%
  14700  5522      505.5463    25      611.0516     441.9200  1749806   27.68%
  15500  5486    infeasible            611.0516     445.2233  1842569   27.14%
  17000  5767      457.1799    93      611.0516     450.6236  2027327   26.25%
```

Table 9 Continued

```
        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer     Best Node    ItCnt     Gap

  17100  5823    infeasible            611.0516      450.7417  2035600   26.24%
  18000  5607      585.5001    62      611.0516      455.1719  2144508   25.51%
  19000  5732    infeasible            611.0516      459.4194  2249067   24.81%
  20700  6003        cutoff            611.0516      464.6939  2391293   23.95%
  22000  6050    infeasible            611.0516      470.5530  2558174   22.99%
  24500  5920        cutoff            611.0516      482.3856  2729101   21.06%
  24600  5888      498.9482    50      611.0516      483.5107  2736206   20.87%
  25200  5828      495.4719    80      611.0516      489.2547  2775632   19.93%
  27000  6081    infeasible            611.0516      495.2470  2887185   18.95%
  29300  6307        cutoff            611.0516      506.8384  3043697   17.05%
  29400  6294      507.3607    74      611.0516      507.2671  3052718   16.98%
  31000  6263        cutoff            611.0516      511.7489  3147269   16.25%
  31700  6287        cutoff            611.0516      514.1824  3193498   15.85%
  33100  6438    infeasible            611.0516      519.3399  3292744   15.01%
  33200  6476        cutoff            611.0516      519.4526  3299353   14.99%
  36000  6949      541.5859    58      611.0516      525.4215  3468670   14.01%
  39100  7502        cutoff            611.0516      531.1301  3623554   13.08%
  42600  7667        cutoff            611.0516      536.7354  3811924   12.16%
  48200  7694    infeasible            611.0516      543.7067  4085500   11.02%
  48400  7656        cutoff            611.0516      543.9900  4095403   10.97%
  53100  7984      550.4803    35      611.0516      549.8545  4300963   10.02%
  58100  7854      556.4642    47      611.0516      556.0538  4505492    9.00%
  63700  7416    infeasible            611.0516      562.1809  4687208    8.00%
  68800  6609    infeasible            611.0516      568.2867  4839602    7.00%
  74000  6264        cutoff            611.0516      574.1363  4994964    6.04%
  78800  5751    infeasible            611.0516      579.9609  5114362    5.09%
  79200  5723      580.4655    59      611.0516      580.4389  5122304    5.01%
  83300  4779    infeasible            611.0516      586.4079  5205738    4.03%
  87000  3994      592.8709    41      611.0516      592.0453  5273957    3.11%
  90700  2641        cutoff            611.0516      599.4204  5337221    1.90%
  93600  1600    infeasible            611.0516      604.8722  5373857    1.01%
  95500   822    infeasible            611.0516      607.9319  5391712    0.51%
  96500   413        cutoff            611.0516      609.5331  5400660    0.25%
  96800   197        cutoff            611.0516      610.2933  5402815    0.12%
  97000    53        cutoff            611.0516      610.8933  5404603    0.03%
Elapsed b&b time = 2593.59 sec. (tree size =  0.16 MB)

Flow cuts applied:  4
CPLEX 6.5.2: optimal integer solution within mipgap or absmipgap; objective 611.
0515933
5404660 MIP simplex iterations
97033 branch-and-bound nodes
2 simplex iterations (0 in phase I)
ampl:
```

APPENDIX C

Table 10
Output results from routing system

TRUCK ROUTES

```
Travelspeed : 40 MPH                              Truck Time   :12 Hrs
Unloadtime  : .5 Hrs                              Truck Weight:48000 Lbs

Route Number 1
Location                                    Arrival Time  Departure Time
----------------------------------------------------------------------
HomeDepot                                   -             06.10
029-Lowes 551 @ Lewisville                  07.00         07.30
029-All Texas Fence @ Lake Dallas           07.40         08.30
029-Barksdale Fence Supply Co. @ Argyle     08.45         09.15
029-Lowes 550 @ Carrollton                  09.50         10.20
029-Lowes 1059 @ Frisco                     10.40         11.10
029-Lowes 1199 @ Allen                      11.25         11.55
029-Lowes 611 @ Garland                     12.15         12.45
029-Fence Parts, Inc. @ Ft. Worth           14.00         14.30
029-Lowes 525 @ Ft. Worth                   14.40         15.10
----------------------------------------------------------------------
No. Of Locations = 9
Total Distance = 213.2 miles
Total Weight = 45602 lbs
Total Volume = 88159
Total Time = 10.05 hours

Route Number 2
Location                                    Arrival Time  Departure Time
----------------------------------------------------------------------
HomeDepot                                   -             09.10
029-Lowes 520 @ S. Arlington                10.00         10.30
029-Dalworthington Fence @ Arlington        10.35         11.05
Master-Halco, Inc./Arlington                11.15         11.45
Master Halco, Inc./Ft. Worth                12.15         12.45
029-Hurricane Fence Company @ Denton        13.45         14.15
029-Allied Fence of Sherman @ Sherman       15.55         16.25
----------------------------------------------------------------------
No. Of Locations = 6
Total Distance = 236.2 miles
Total Weight = 46292 lbs
Total Volume = 26551
Total Time = 8.50 hours
```

Table 10 Continued

```
Route Number 3
Location                                    Arrival Time  Departure Time
----------------------------------------------------------------------
HomeDepot                                   -             04.35
029-Cen-Tex Fence Company, Ltd. @ Temple    08.00         08.30
029-Super Tramp aka Acme Fence @ Waco       09.20         09.50
029-Arrow Fence Co. @ Waco                  09.50         10.20
029-Brem's Fencing & Repair, Inc. @ Waco    10.35         11.05
029-Mid-Tex Fence & Repair @ Elm Mott       11.10         11.40
029-Devco Fence and Supply @ Waco           12.20         12.50
029-PWS Renovations @ Waxahachie            14.50         15.20
----------------------------------------------------------------------
No. Of Locations = 7
Total Distance = 326.9 miles
Total Weight = 32570 lbs
Total Volume = 45985
Total Time = 11.40 hours

Route Number 4
Location                                    Arrival Time  Departure Time
----------------------------------------------------------------------
HomeDepot                                   -             05.10
029-FOXWORTH 25 @ PARIS                     08.00         08.30
029-Emerson Fence Co. @ Paris               08.35         09.05
029-Metro Gate & Manufacturing @ Paris      09.25         09.55
029-Foxworth 42 @ Sulpher Springs           10.40         11.10
029-East Texas Landscaping @ Sulphur Spr    11.15         11.45
029-Nathaniel Fence @ Sulphur Springs       11.50         12.20
----------------------------------------------------------------------
No. Of Locations = 6
Total Distance = 239.8 miles
Total Weight = 43328 lbs
Total Volume = 19204
Total Time = 8.55 hours

Route Number 5
Location                                    Arrival Time  Departure Time
----------------------------------------------------------------------
HomeDepot                                   -             07.15
029-Diamond B Fence Company @ Arlington     08.00         08.30
----------------------------------------------------------------------
No. Of Locations = 1
Total Distance = 55.8 miles
Total Weight = 45806 lbs
Total Volume = 9967
Total Time = 1.50 hours
```

Table 10 Continued

```
Route Number 6
Location                                    Arrival Time  Departure Time
-----------------------------------------------------------------------
HomeDepot                                   -             07.25
029-North Texas Crown Fence Co. @ Wylie     08.00         08.30
-----------------------------------------------------------------------
No. Of Locations = 1
Total Distance = 41 miles
Total Weight = 45461 lbs
Total Volume = 32080
Total Time = 1.30 hours

----------------------------------------
Total distance: 1112.9 miles
----------------------------------------


-----------------------------------------------------------------------
Total Cost = (1112.9 * .7)+(6 * 8 * 15)+(7.715 * 15 * 1.5) = $ 1672.61
-----------------------------------------------------------------------
```

VITA

Gopalakrishnan Easwaran
1. P. N. Street, S. S. Colony,
Madurai - 625 010.
Tamil Nadu, India.

Education

Master of Science, Industrial Engineering August 2003
Department of Industrial Engineering
Texas A&M University, College Station, Texas, USA.
Thesis Topic: Design and development of a vehicle routing system under capacity, time-windows and rush-order reloading considerations.

Bachelor of Engineering, Mechanical Engineering May 2000
Department of Mechanical Engineering
PSG College of Technology, Coimbatore, Tamil Nadu, India.

Work Experience

Graduate Assistant - Research, Industrial Distribution, TAMU. (May 2001 - present)
Coordinated short term consultancy projects involving collaborative forecasting, aggregate planning, network optimization, and warehouse layout designs for manufacturing and service companies in and around Texas.

Student Technician, Purchasing Services, TAMU. (Feb 2001 - May 2001)
Member of the System Design Team at Purchasing Services and involved in developing an inventory system for efficient warehouse management using MS SQL Server and Visual Basic programming.