J. Humanities (2007) Vol. 14 (1): (27 - 41)

The Decision Support System for the Resolution of Real-Life Vehicle Routing and Scheduling Problem

Babak Sohrabi[1*], Babak Abedin[2]

Received: 30/5/2005 Accept: 17/7/2006

Abstract

In this paper, we investigate a decision support system (DSS) for the resolution of real-life vehicle routing and scheduling problem (VRSP). Scheduling the deliveries from a regional distribution centre (RDC) to large stores of a major fmcg retailer includes every possible vehicle routing complexity. Usual constraints that are seen are: size of the vehicle and the length of the driving day, loading feasibility of products in different parts of the vehicle, and also with various time windows. More importantly, in this scheduling descion-making is customer oriented, in which, customer's value for the company is considered as one of the most important factors. The algorithms for the resolution of the distribution problems constitute a very important part of DSS. Therefore, a simulated annealing based algorithm has been developed to speed up the process by circumventing the need for the skeletal schedule.

Keywords: Decision Support System (DSS), Vehicle Routing and Scheduling Problem (VRSP), Simulated Annealing (SA)

*1. Dept of Information Technology Management, Faculty of Management, University of Tehran, Tehran, Iran, E-mail: bsohrabi@ut.ac.ir

2. M.Sc. Information Technology Management,University of Tehran, Tehran, Iran, E-mail: B_abedin@mail.iust.ac.ir

1. Introduction

A DSS is a computer based system that has the function of assisting the decision maker in solving difficult problems and taking the most appropriate decisions [1]. As such, there is increasing recognition that effective decision-making requires the incorporation of vehicle routing techniques into a decision support system (DSS): User intervention is often the only way of ensuring that the routs meet "soft" constraints that are not part of the mathematical formulation. Experienced schedulers have considerable knowledge of local conditions and relative importance of the various constraints. Also, human intervention in the routing process can substantially be the quality of routs produced [2].

In practice, there are many factors that make the VRSP a very complex problem [3]. The first difficulty is to find out among the existing algorithms is as to what the appropriate action be taken to solve a given VRSP. Similarly, for a given algorithm and VRSP, what are the best choice of parameters. Another difficulty is the dynamism of the data. The orders, though change every day but, there are many other data that change more or less frequently: availability of the vehicles and drivers, new calls and changes in the characteristics of the old ones, new road or closure of the old ones; different traveling costs due to changes in the wages, fuel cost, vehicles cost, etc. The orders to be delivered every day may not be well defined, if the customers do not specify the delivery date. A DSS may help in all those difficulties, and thus for a planner who is neither an expert in OR nor in computing science, it must prove user friendly. The most obvious aspect is that DSS can be useful in the management of the database. Further, in the resolution process it can also play an important role. When the VRSP resolution algorithm depends on parameters, the DSS can help in the choice of the parameters automatically, or can be used to calculate the costs of different solutions with different parameters.

In addition to above constraints, customer's value is considered as a very impressive factor in vehicle allocation to stores. It means that every customer has it's own value and historical experiences with company and also their current status in market is important in decision making and assigning a travel to them.

2. The Vehicle Routing and Scheduling Problem

2. 1 Background

Routing problems typically have a requirement for customers at known locations, to be supplied with their demand by a set of vehicles; subject to limitations on the capacity of the vehicles, the duration of the routes, and the time at which the customer receives a delivery [2]. In mid-1960s, VRP solutions were based on non-computerized methods, where routes were designed manually by using paper maps that provide locations of the customers to be routed as geographical reference points. By the early 1970s, vehicle routing heuristics were used to produce routes, together with skilled manual intervention by experienced decision makers who had considerable knowledge of local condition and relative importance of the various constraints. From the early 1980s, every vehicle routing systems was seen as a decision support system (DSS) [4] and use of decision support systems is VRP problems increased significantly [5,6].

2. 2 Description of Real World Problem

Developments in vehicle routing have been driven by the need to obtain solutions to ever more complex real-world problems and then to improve upon those solutions. Thus, additional complications, like having multiple depots and multi-size vehicles and the logistics problem faced by the UK's major superstore operators; Asda, Safeway, Sainsbury and Tesco, have been considered. Now, increasingly selling durables, have evolved from grocery retailers, with operations as described below.

Suppliers do not normally deliver directly to any stores rather each must deliver to a regional distribution centre (RDC). However, not all RDCs will stock every product. Then the basic problem arises is the schedule deliveries from a RDC to the retail stores in its area, if necessary picking up goods end route from a central warehouse, generally referred to as a hub depot.

Whilst a superstore retailer will stock thousands of products, in bar code terms, there are basically five product categories for distribution.

Grocery items only stocked at a hub depot, from where deliveries must be made directly to retail outlets, *Grocery* items available at a RDC for delivery to the stores, *Produce*, as sold by the traditional greengrocer, *Perishable* goods, like dairy products, *Frozen* products, which are predominantly foods.

A pallet may contain different products from the same category but products from different categories will not be on the same pallet. The delivery scheduler is only interested in the number of pallets of each category required by a store and is unconcerned about the mixed product on the pallets. Details about the vehicle loading are shown in Figure 1.

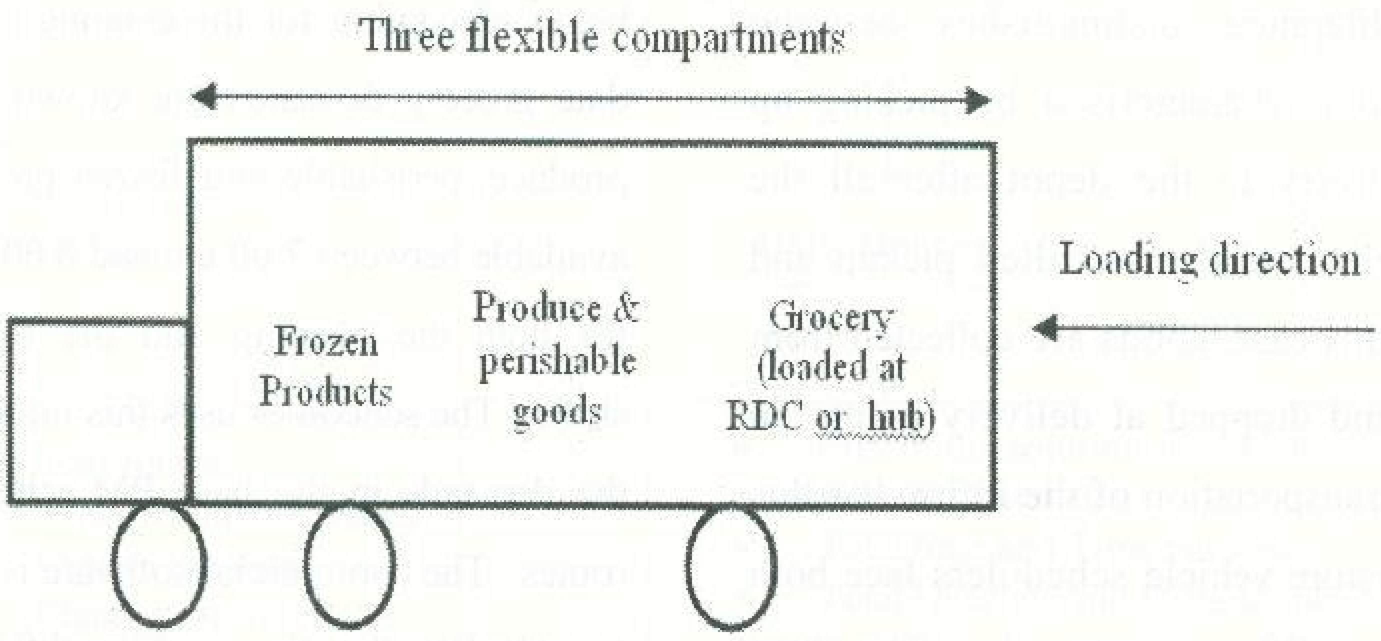

Figure 1 As the sizes of the compartments are totally flexible, a full vehicle could carry one, two or all three of these product types

Rear end loading and unloading precludes total flexibility, in terms of the mix of product types carried by a vehicle, if it is to deliver to more than one store. For example, a vehicle would have to deliver grocery to the first store, produce or perishables to the second store and then return to the first store with the frozen products. Similarly, a vehicle partly laden with pallets from the RDC could not travel to the hub and fill up with pallets from there, if the first delivery was for only RDC loaded goods. Thus, the product type mix of pallets for the various stores is a significant determinant of the vehicle delivery schedule, not covered in the VRP literature.

Because of their importance, time window was the first complexity introduced into the VRP. In practice, there are both hard and soft time windows, with hard time windows not to be violated at any cost. However, soft time windows allow a vehicle to arrive early or late but a penalty, depending on the degree of violation, is imposed it.

The VRP literature distinguishes between backhauls, which is characterised by picking up products for delivery to the depot after all the outgoing deliveries and the so-called pickup and delivery [7]. In this case, goods are collected from pick-up points and dropped at delivery points on the same route; transportation of the infirm has this structure. Superstore vehicle schedulers face both these problems at different intervals. The most frequent type of collection are from stores to any depot of salvage, which is packaging materials, pallets etc. This can only be possible when all the product deliveries are made. Products are also collected from suppliers, usually for delivery to the depot where the vehicle is based, but more complex backhauls are possible. For example, a collection may be made from a supplier, after delivering to a nearby store, and trunked to a distant depot, before returning to its base, making an inter-depot transfer in the process.

As well as the limit to a vehicle's pallet capacity, the driver's time is also regulated, in terms of both the length of the driving day and rest periods. The vehicle schedules must include allowances for loading and unloading, depending on the number of pallets involved and preparation times, which are independent of the number of product categories.

The collaborating retailer uses a commercially available routing and scheduling package to produce a base schedule for morning (AM) and afternoon (PM) deliveries every day of the week. The base schedule is updated every six months, because the software is not designed to handle complex routing and is insufficiently fast to route, obtained for each shift. The scheduler at a depot arrives at 6.00 am to begin scheduling for the evening deliveries. Up-to-date grocery demands are known, whilst those for produce, perishable and frozen products are usually available between 7.00 am and 8.00 am, as backhauls for both the evening and the following morning shifts. The scheduler uses this information to update the demands in the base PM schedule and modify routes. The commercial software is used at this stage to calculate the times of modified routes and to prevent the scheduler violating either the vehicle capacity or hard time window constraints. Obviously, if some demands have been increased, then the scheduler may be unable to make complex adjustments, resulting in split-loads. This process is completed by 9.30 am and the printed schedule forwarded to the warehouse for loading. At 10 am, the scheduler starts the same process for the next day's morning schedule. Nevertheless, a schedule is printed at 1 pm and retained until 6 pm, when the required grocery data arrives. The AM schedule is then updated manually and passed to the warehouse for loading the vehicles, which will leave the RDC at regular intervals from 3.30 am onwards.

Grocery's demand has a significant share of total demand. New orders of grocery receive every day between 1pm to 6pm, but other types of product have predictable and fix quantities very day. Therefore, these new orders influence the loading scheduling. There is a customer-ranking model, and then accordingly, vehicles loading priority are arranged.

3. Design of the DSS

DSS has four main modules, which are *data management subsystem*, *model management subsystem*, *knowledge management subsystem* and *user interface subsystem* [8]. According to user's need, a simple and flexible DSS is developed in vehicle routing and scheduling problem. It is programmed by C language because C is an efficient and portable language. Also, as C is structured language, the maintenance and future changes will be easy, without many changes.

The function scheme of the DSS is illustrated in Figure 2. The system receives the data from two databases: static and dynamic .

In the next sections, different parts of the DSS are described.

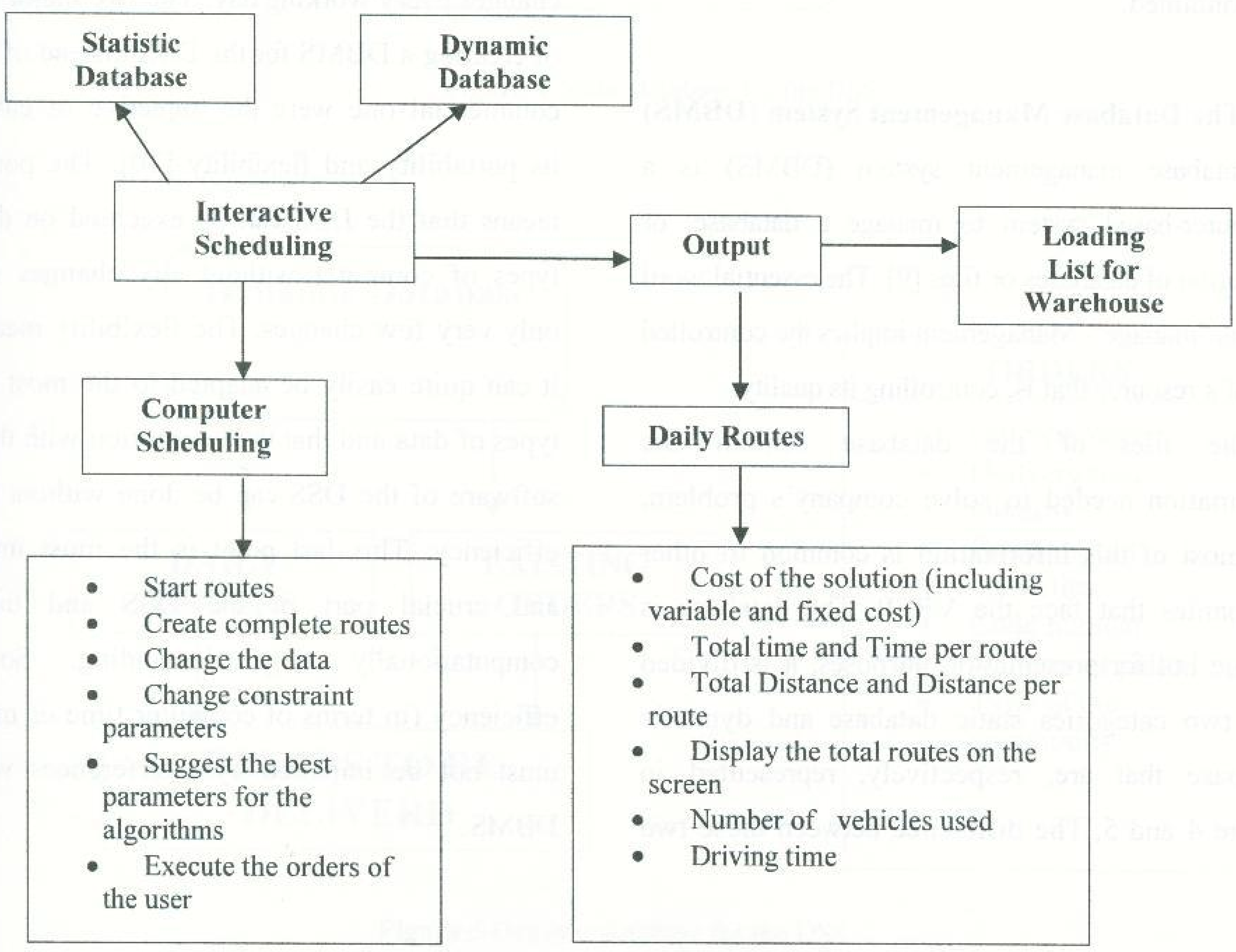

Figure 2 Function representation of the DSS

3. 1 A Simple User Interface

To make it easy and understandable, the design of user interface is based on traditional process of decision making in the company. The process of data entry is question-based loop, in which includes the customer and products' details, time of delivery, demand, and other information. A schematic view of the process is presented in Figure 3.

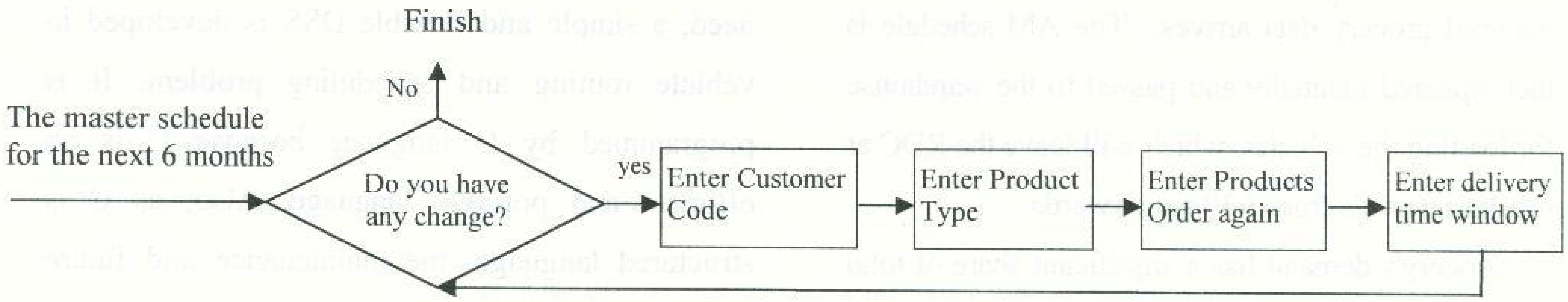

Figure 3 A schematic view of process of data entry

The delivery time is entered in a " from … to…" format and customers are defined by unique codes, in which the routes to them and their details are contained.

3. 2 The Database Management System (DBMS)

A database management system (DBMS) is a computer-based system to manage a database, or collection of databases or files [9]. The essential word here is "manage". Management implies the controlled use of a resource that is, controlling its quality.

The files of the database contain the information needed to solve company's problem, but most of this information is common to other companies that face the VRSP. The database is unique but for presentation purposes, it is divided into two categories static database and dynamic database that are, respectively, represented in Figure 4 and 5. The difference between these two databases is that while the first usually does not see change from one day to another or experiences only a few changes, the latter has more data changes every working day. The two major reasons of creating a DBMS for the DSS instead of using a commercial one were the objective of enhancing its portability and flexibility [10]. The portability means that the DSS can be executed on different types of computer without any changes or with only very few changes. The flexibility means that it can quite easily be adapted to the most diverse types of data and that the integration with the other software of the DSS can be done without loosing efficiency. This last point is the most important and crucial part of the DSS and they are computationally very demanding. So their efficiency (in terms of computer time or memory) must not be impaired by interferences with the DBMS.

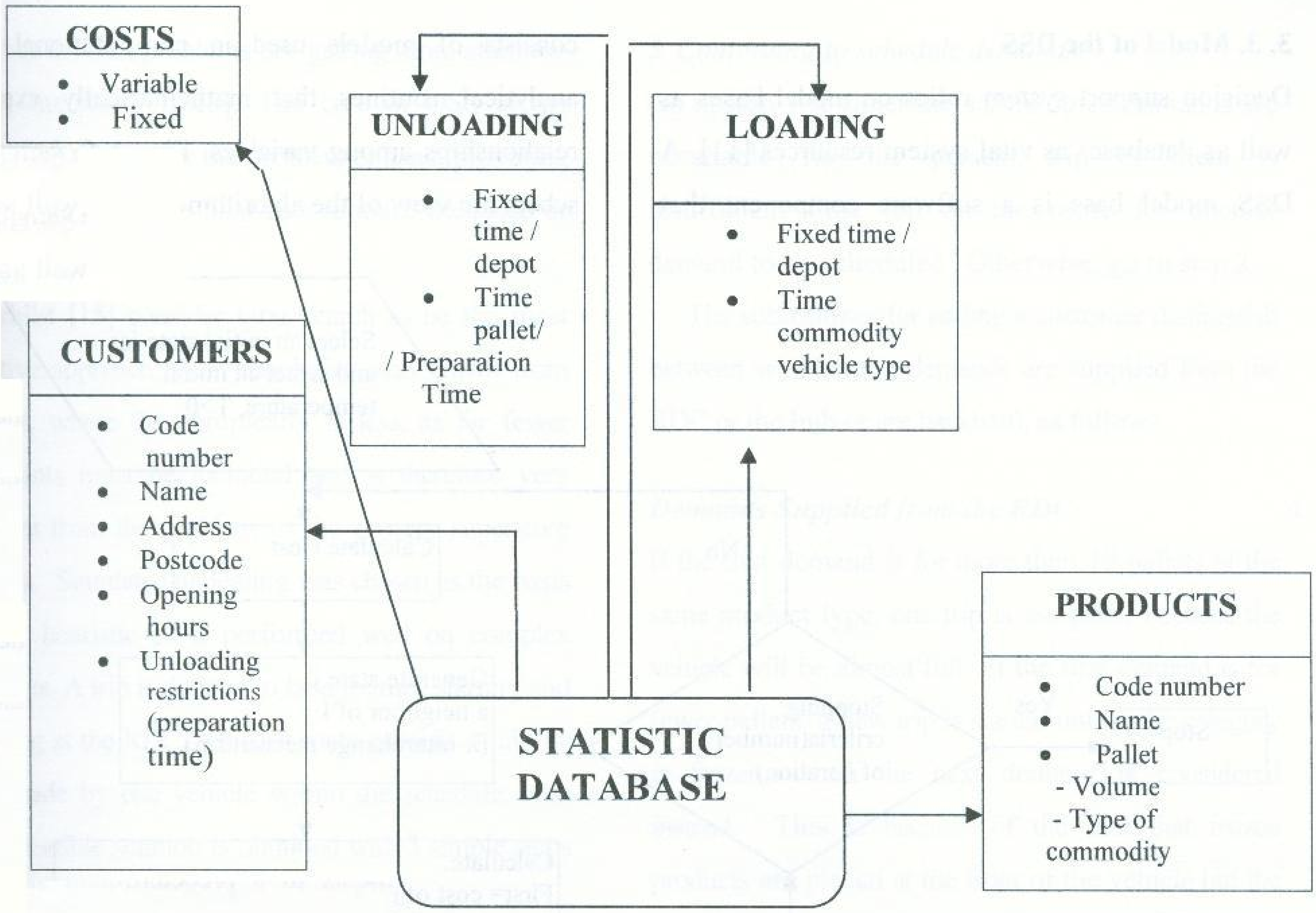

Figure 4 Static database for the DSS

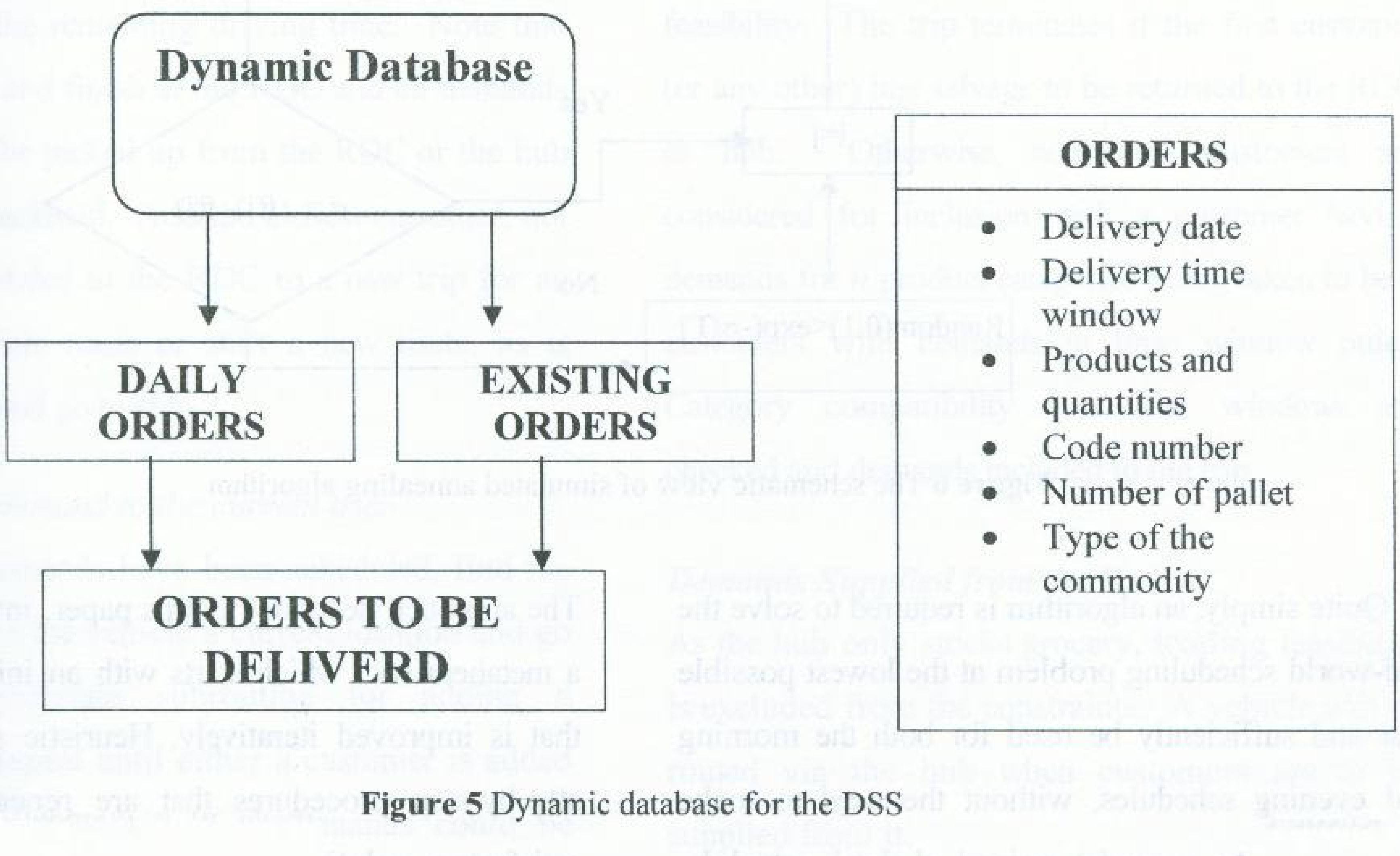

Figure 5 Dynamic database for the DSS

3. 3. Model of the DSS

Decision support system relies on model bases as well as databases as vital system resources [11]. A DSS model base is a software component that consists of models used in computational and analytical routines that mathematically express relationships among variables. Figure 6 shows the schematic view of the algorithm.

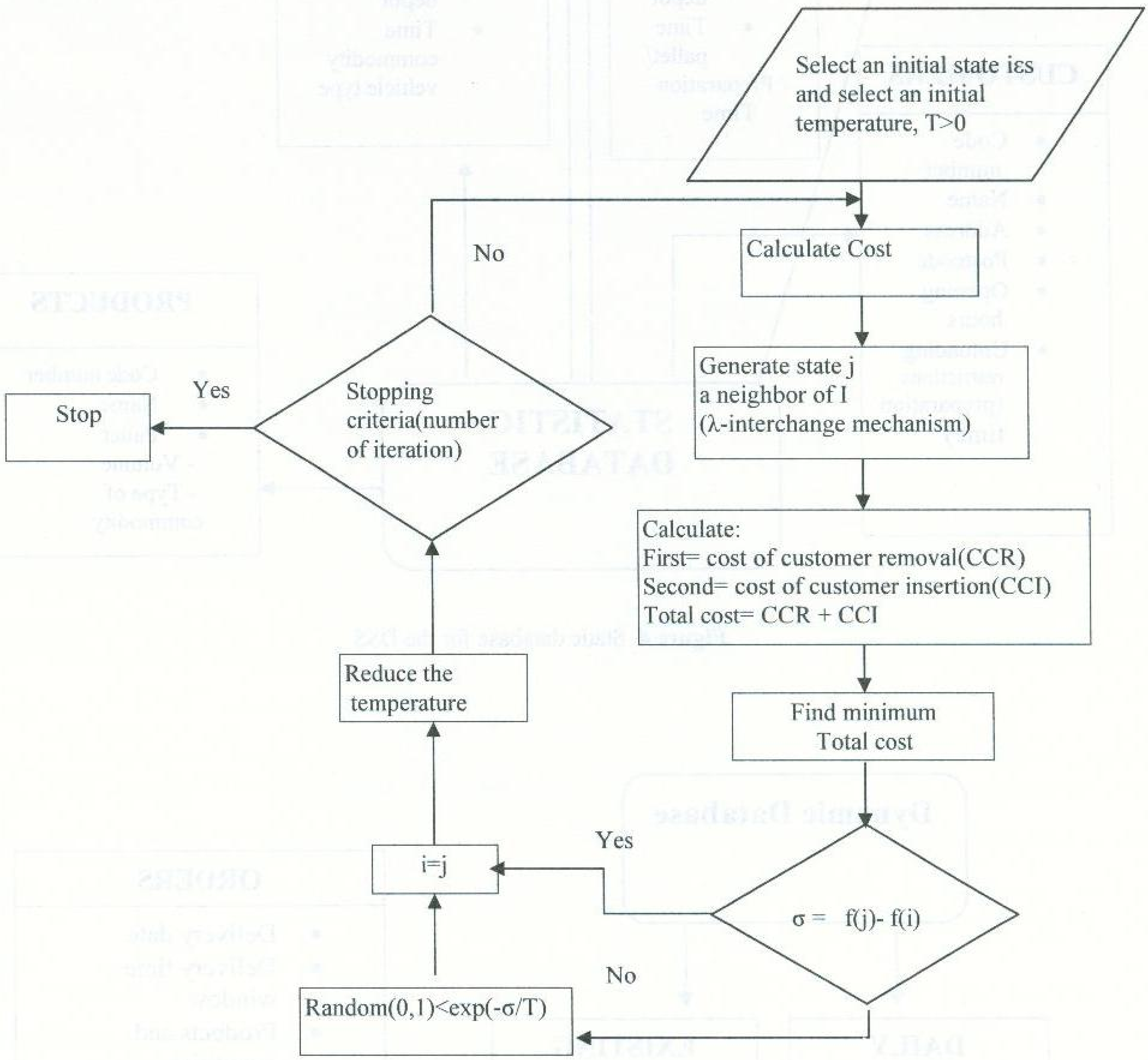

Figure 6 The schematic view of simulated annealing algorithm

Quite simply, an algorithm is required to solve the real-world scheduling problem at the lowest possible cost and sufficiently be used for both the morning and evening schedules, without the need to make adjustments to a predetermined skeletal schedule. The approach described in this paper, involves using a metaheuristic, which starts with an initial solution that is improved iteratively. Heuristic searches are step-by-step procedures that are repeated until a satisfactory solution is found [10]. Use of heuristic searches in VRP problems are getting more attention in the literature [12,13]. The core of DSS is the heuristics which solve the problem at hand, providing output that allows the final user to make decisions or perform analysis [14].

Whilst [15] consider tabu search to be the most effective approach, their conclusion was drawn from CVRPs, where the complexity is less, as far fewer constraints must be included and is therefore very different from the problem of the grocery superstore retailers. Simulated annealing was chosen as the basis for the heuristic as it performed well on complex problems. A trip is defined to be a journey starting and finishing at the RDC, whilst a route consists of all the trips made by one vehicle within the schedule. An initial feasible solution is obtained with 3 simple steps that construct vehicle routes sequentially.

1. Starting a new trip or route.

If there are still demands to be met, whether a new trip within the present route or a new route is started depends on the remaining driving time. Note that all trips start and finish at the RDC and all demands are either to be picked up from the RDC or the hub or they are backhaul. Add the closest customer, not already scheduled to the RDC to a new trip for an existing vehicle route or start a new route, as is appropriate, and go to step 2.

2. Adding a demand to the current trip.

Unless all demands have been scheduled, find the one nearest to the vehicle's current location and go to the appropriate subroutine for adding a customer. Repeat until either a customer is added or none of the unscheduled demands could be included in the trip. Then go to step 3.

3. Continuing to schedule demands.

Go to step 1 if either the unscheduled demands are unsuitable for the present trip or there is insufficient time left on the route for another demand to be scheduled. Otherwise, go to step 2.

The subroutines for adding a customer distinguish between whether the demands are supplied from the RDC or the hub or are backhaul, as follows.

Demands Supplied from the RDC

If the first demand is for more than 19 pallets of the same product type, one trip is assigned, because the vehicle will be almost full. If the first demand is for fewer pallets, a new trip is started unless the category is frozen, when the next demand is considered instead. This is because of the fact that frozen products are placed at the front of the vehicle but the first scheduled demand is at the rear. When the first customer requires more than one category, all those with overlapping time windows are considered. The capacity and driving time constraints are checked for feasibility. The trip terminates if the first customer (or any other) has salvage to be returned to the RDC or hub. Otherwise, additional customers are considered for inclusion with a customer having demands for *n* product categories being taken to be *n* customers with demands in time window order. Category compatibility and time windows are checked and demands included in the trip.

Demands Supplied from the Hub

As the hub only stocks grocery, loading feasibility is excluded from the constraints. A vehicle will be routed via the hub when customers are to be supplied from it.

Backhauls

The time window of a backhaul is checked before its insertion in a trip. When the backhaul is delivered to the RDC, a new trip may be started. If it is delivered elsewhere, such as to another supplier or RDC, and if sufficient driving time is available, then the trip may be continued by returning to step 2.

In obtaining an initial solution, the number of vehicles is not limited but hard time windows are used. The improvement algorithm is essentially standard simulated annealing, except that a given number (M) of best-improve moves are made at each of the (N) iterations with the same temperature, before consideration is given to the probabilistic acceptance of a worse solution. Following [16], the λ-interchange generation mechanism is used to explore the neighbourhood structure. For deliveries from the RDC, λ = 1 is used, so that two customers in different trips may be swapped or a customer may simply be shifted from one trip to another. However for deliveries from the hub, the pick-up from the hub and the delivery of those goods are not allowed to be moved independently (i.e. λ = 2), because the hub and the customer requiring the hub groceries must be in the same trip. Similarly for a backhaul, both its origin and destination must moved together. The simulated annealing stops when a prescribed number of cycles of N iterations are executed without changing the value of the best solution.

The costs of distribution, which change with the schedule, are the tractor, trailer and driver fixed costs and the variable cost per mile. Thus when a customer is removed from a route, there is a cost reduction equal to that for one vehicle, if the route contained one customer, and zero otherwise. In addition, there will be a decrease in the mileage cost. The converse applies when a customer is added to a route.

In obtaining an initial solution, it is assumed that the vehicle capacity is fixed. In the real world, the scheduler can always load a few extra pallets on to a nominally full vehicle and the time windows are soft. Hence the improvement algorithm includes penalty costs for overloading, expressed as pounds per additional pallet, and time window penalties in pounds per hour waiting or late at a customer. Any changes in these penalty costs must be included with the vehicle and mileage cost changes, whenever a customer is removed from or added to a route. Obviously the change in the capacity penalty, which is usually zero, depends on the vehicle capacity and the number of pallets on the vehicle. However, the removal or addition of a customer may alter the time window penalties for later customers in the route.

When a customer is added to a trip, loading feasibility has to be checked against the product categories of the previous and following customer. This is unnecessary for a removal.

A trip with only one customer is not destroyed if that customer is removed during the simulated annealing, because it provides an opportunity to insert another customer. In practice, it is expected that the improvement algorithm will reduce the number of vehicles, unless the initial solution is very good. The λ-interchange mechanism could be extended but it becomes complex for sets of customers. Instead, it can very easily be applied to complete trips in routes, instead of customers, with $\lambda = 1$. No costs will change, except the time window penalties, and the only constraint is the length of the driver's day.

4. The Application Overview

The collaborating company services 108 retail outlets from one RDC, as well as collecting from suppliers and trunking between depots. Table 1 gives details of one particular schedule, when 15 vehicles moved 538 pallets, excluding trunking. As table shows an initial solution method that provides an initial feasible solution (15 routs) to the problem and which is improved (12 routs) by a novel from of SA algorithm.

Table 1 Summary routing statistics

	Initial solution	Final solution
Routes	15	12
Trips	32	22

The output of program same as follow, but in this paper, just route 1 is showed as a sample of the application output:

Notation: under action, D= deliver, P= pickup.

```
Name          Type          dprt   arvl  volume action route trip

warrington    --            5.31   0.00   0       -      1     1
middleton     BH            6.41   6.14   5       P      1     1
------------------------------------------------
warrington    --            7.58   7.24   5       D      1     2
chorleton     WARN PERI     8.58   8.33   6       D      1     2
trafford_p    BH            10.08  9.08   21      P      1     2
------------------------------------------------
warrington    --            11.13  10.39  21      D      1     3
swinton       TAMW/TH GROC  12.09  11.43  9       D      1     3
west_kirby    TAMW/TH GROC  14.00  13.38  5       D      1     3
------------------------------------------------
warrington    --            15.46  15.12  0       -      1     4
upton         WARN/TN GROC  16.46  16.29  2       D      1     4
upton         SALVAGE       16.46  16.46  21      P      1     4
warrington    --            0.00   17.29  0       -      1     4

Time     = 10.58 hors
Distance = 269.00 miles
Pallet number  War =22  Tam= 0
```

The sequence of events, which characterises the first rout, is:

1. Leave at 5:31 and travel empty to Middleton, arriving at 6:14;

2. Leave Middleton at 6.41 to backhaul 5 pallets to Warrington, arriving at 7:24;

3. Leave Warrington for second trip at 7:58 with 6 pallets of perishable for chorleton, arriving at 8:33;

4. Leave Chorleton at 8:58, arriving at Trafford Park at 9:08;

5. Leave Trafford Park at 10:08 to backhaul 21 pallets to Warrington, arriving at 10:39;

6. Leave Warrington for third trip at 11:13 with 9 pallets of grocery for Swinton and 5 pallets of grocery for west Kirby;

7. Arriving at Swinton at 11:43 to deliver 5 pallets of grocery and leaving at 12:09;

8. Arriving at West Kirby at 13:38 to deliver 9 pallets of grocery and leaving at 14:00;

9. Arriving at Warrington at 15:12;

10. Leave Warrington for fourth trip at 15:46 to deliver 2 pallets of grocery to upton and pickup 21 pallets of salvage from there;

11. Arriving at uptown to deliver 2 pallets grocery at 16:29 and picking up 21 pallets of salvage. Although we allow up to 21 pallets of salvage, it is often less than that in practice and does not take long to road;

12. Leave Upton at 16:46 with 21 pallets of salvage , arriving Warrington at 17:29.

Figure 7 shows the map of explained route.

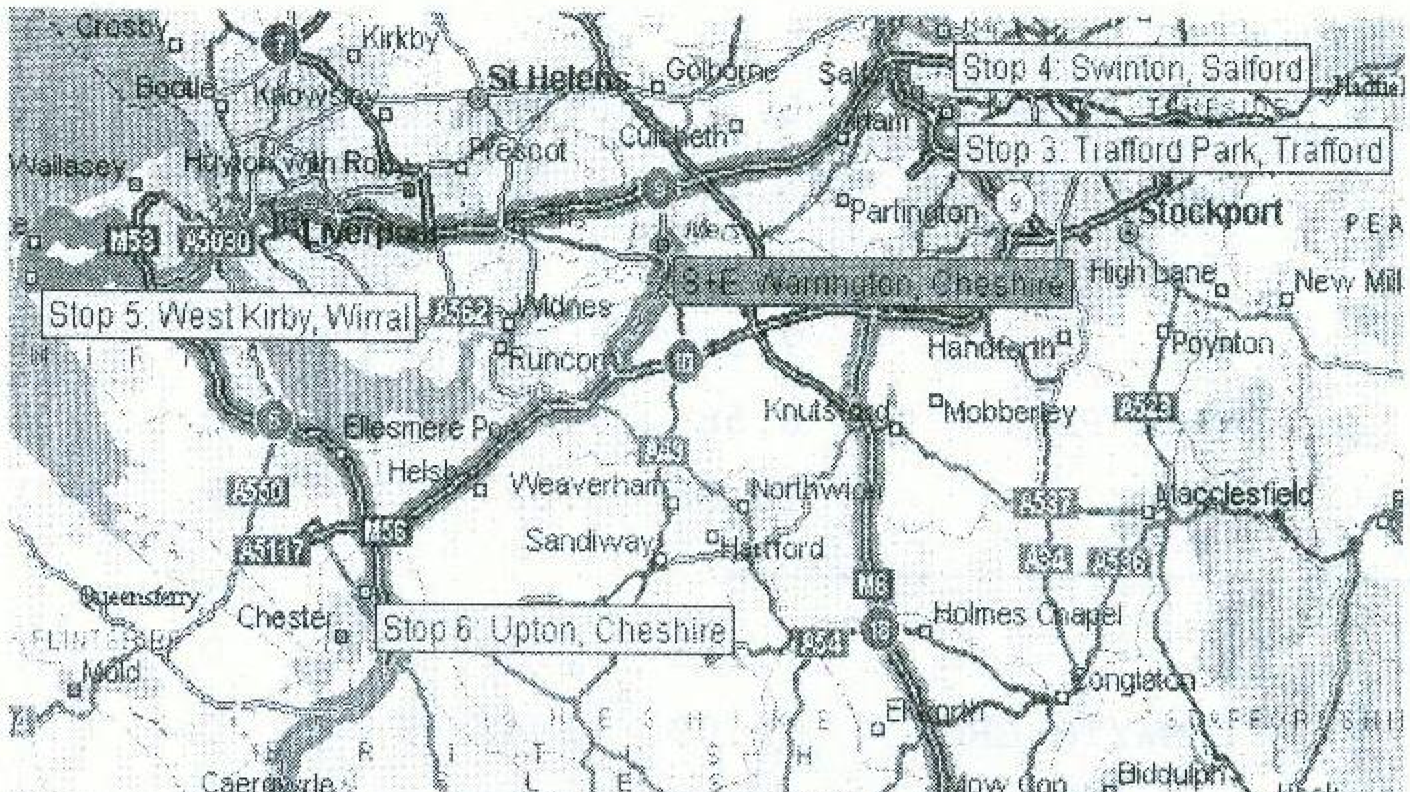

Figure 7 The map or route 1 (the black box shows RDC)

5. Conclusion

In this paper a decision support system (DSS) was created for solving the distribution problem, which is typically faced by companies operating supermarkets and other types of shop. The functions of the DSS are the following: to make communication easier between the user and the database, and between the database and the algorithms; to improve the interface between the algorithms. The algorithm for the resolution of the distribution problems constitutes a very important part of DSS. The algorithm is able to produce routes sufficient quick to obviate the need for a base schedule that is modified daily, with input from a very experienced distribution professional. Not very good initial solution is modified by a simulated annealing heuristic to produce the routes. Most importantly, the scheduler uses knowledge of the specific problem, whereas the metaheuristic is essentially general purpose, with problem-specific constraints. As company tries to keep it's important customers satisfied, one of these constraints is customer satisfaction level. Therefore, according to new demands for grocery during the day and also with regard to limitation of resources, vehicles loading are scheduled-based on customer-ranking model.

References

[1] Sprague, R. H, and carlson, E. D, *Building effective decision support system*; Prentice-Hall, INC., Englewood cliffs, New Jersey, (1982).

[2] Peter B Keenan. *Spatial decision support systems for vehicle routing.* Decision support systems, Vol. 22, pp 65-71, (1998).

[3] Golden, B. L and Assad A. A, *Vehicle Routing: Methods and Studies*, Elsevier Science Publishers B.V, (1988).

[4] C. D. Tarantilis, C. T. Kiranoudis, Using a spatial decision support system for solving the vehicle routing problem, *Information and management*. Vol. 39, pp 359-375, (2002).

[5] G. Zografos Konstantinos, N. Androutsopoulos Konstantinos, A heuristic algorithm for solving hazardous materials distribution problems, *European Journal of perational Research* 152, 507–519, (2004).

[6] Rodriguez' Patricio, Nussbaum Miguel, Baeza Rodrigo, Leon Gerardo, Sepdlveda Marcos, Cobib Agustin, Using global search heuristics for the capacity vehicle routing problem, *Computers Operation Research*,Vol 25, No. 5, pp: 407-417, (1997).

[7] Goetschalckx M and Jacobs-Blecha C, The vehicle routing problem with backhauls, *European Journal of Operatinal Research*, Vol. 42, pp39-51, (1989).

[8] Efraim Turban, Jay E Arondon, *Decision support system and intelligent systems*, Fifth edition, Prentice-Hall, New jersey, (1998).

[9] Gordan C Everest. *Database management objectives, system Function, and administration*, McGraw Hill, (1986).

[10] AI Stevens, *C Database Development*, Management Information Source, Inc, (1991).

[11] James O'Brien. *Introduction to information systems*, Tenth edition, McGraw-Hill, (2001).

[12] Fagerholt Kjetil , A computer-based decision support system for vessel fleet scheduling-

experience and future research. *Decision Support Systems*, Vol. 37, pp 35– 47, (2004).

[13] Ruiz Ruben, Maroto Concepcion, Alcaraz Javier, A decision support system for a real vehicle routing problem, *European Journal of Operational Research*, Vol. 153, pp 593–606, (2004).

[14] Patricio Rodriguez, Miguel Nussbaum, Rodrigo Baeza, Gerardo Leon, Marcos Sepulveda, Agustin Cobian, Using global search heuristics for the capacity vehicle routing problem, *Computers Ops*. Vol 25, No 5, pp 407-417, (1998).

[15] Gendreau M, Laporte G and Potvin J-Y, Metaheuristics for the capacitated VRP, In: Toth P and Vigo D (eds). *The vehicle routing problem,* SIAM Mongraphs, Philadelphia, PA; pp 129-154, (2002).

[16] Osman I H, Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, *Annals of Operations Research*, Vol. 41, pp 421-451,(1993).