A study of the truck dispatching problem using computer simulation

Author:
Whitehead, Graham Henry

Publication Date:
1969

DOI:
https://doi.org/10.26190/unsworks/11866

THE UNIVERSITY OF NEW SOUTH WALES

SCHOOL OF ELECTRICAL ENGINEERING

"A STUDY OF THE TRUCK DISPATCHING PROBLEM USING COMPUTER SIMULATION"

A THESIS PROJECT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING SCEINCE.

Submitted by: Graham Henry Whitehead

Date: February 28th, 1969

Signed:

SUMMARY.

A major portion of the cost of many consumer goods is attributable to the distribution of the goods from the manufacturer to the retailer. This is particularly true where the manufacturer and retailers are located in the one city and distribution is carried out by trucks, each truck making deliveries to several retailers. Several studies have been made to determine the optimum routes that should be followed by the trucks, however, all of these are based on the assumption of a fixed and known demand and ignore the capital cost of the trucks used. It would be extremely difficult, if not impossible, to find analytical solutions to this truck dispatching problem without the assumptions mentioned above, therefore to solve the real problem it is necessary to employ a technique such as simulation.

This paper describes a model developed of a truck dispatching system where the product delivered is in returnable containers owned by the manufacturer and also gives a FORTRAN program that can be used to simulate such a system. The program developed is flexible enough to be used in the study of many problems associated with the dispatching of trucks. At the completion of a simulation the program generates sufficient statistics on the simulation to enable a comparison to be made of the efficiency of various methods of carrying out the dispatching of trucks.

ACKNOWLEDGEMENTS

The author is indebted to Mr. G. McMahon for his guidance and encouragement throughout the project described in this paper.

CONTENTS.

1. INTRODUCTION.

1. Included in the cost of almost all goods is a charge for the distribution of the goods from producer to consumer. This distribution may be by pipeline, truck, 'plane, ship, a combination of any of these or some other method. Whatever the method chosen attention must be given to the cost, as in many cases this greatly effects the economics of an operation. Sometimes there is a choice between several methods of distribution and the problem is to chooose the best alternative; however in most cases there is no alternative method of distribution, but only alternate methods of operating the distribution system, and here the problem is to choose the most efficient method of operation. This is particularly true of the distribution of consumer goods from manufacturer to retailer within the city, where the only method of distribution practical is by motor truck, It is this problem that will be considered in this paper.

In a truck distribution system there are three major sources of cost:-

1. The capital cost of the trucks.
2. The running cost of the trucks.
3. The labour cost of drivers and their "offsiders".

There is also a fourth cost included when the goods are delivered in containers owned by the manufacturer, which are left on the customer's premises until their contents have been consumed, when the containers are returned to the manufacturer for refilling. This is the situation with soft drinks, industrial and medical gases and draught beer.

∴ (4) capital cost of containers.

The aim in studying this problem is to minimize the sum of these four costs while maintaining a "satisfactory level of service" to the customers.

1.1 ANALYTIC APPROACHES.

Many papers have been written giving analytical techniques of optimizing truck distribution systems (Ref. 1, 2, 4, 6, 10 & 13). All of these papers however deal with the problem where the quantity to be delivered to each customer is known and the aim is to minimize the distance travelled.

In a situation where the quantity to be delivered, and the customers requiring deliveries change every day, it is necessary to solve the problem every day (Ref. 13). However this can be both time consuming and costly. Also, these analytical techniques ignore the capital cost of trucks, and where applicable containers.

A study has been made with the purpose of optimizing delivery frequency so as to minimize the sum of the delivery costs and capital costs of containers (unpublished work by the author). However this study used average delivery costs rather than actual costs and assumed a constant rate of consumption by customers.

Both of the above approaches are examples of sub-optimization, that is, a particular area of interest is examined in isolation from all other areas. However, there is always the possibility that an increase in efficiency in one area will cause a decrease in efficiency in another area. Therefore it is desirable to examine the system as a whole, rather than the parts independently. In a system of the complexity of those described here it is extremely difficult if not impossible to formulate analytical solutions, in fact most studies restricted solely to finding optimum routes for known demands cannot be solved exactly and utilize heuristic methods. For these reasons simulation is a logical method of studying the complete system and the interaction of the various parts.

1.2 SIMULATION.

In their book "Computer Simulation Techniques", (Ref. 9 page 3) Naylor et. al. use the following definition of simulation - "Simulation is a numerical technique for conducting experiments on a digital computer, which involves certain types of mathematical and logical models that describe the behaviour of a business or economic system (or some component thereof) over extended periods of real time". This is of course a very narrow definition of simulation, however it is broad enough to cover the term as used in this paper. More simply; in a digital computer simulation study such as that described in this paper a system is hypothesised which has all of the relevant characteristics of the real system being studied, this hypothetical system is the model of the real system. A Computer program is then written which will carry out in chronological sequence, various arithmetic and logical operations which represent operations of the model. As the model is a hypothetical system, it is designed so that it can be represented exactly by a computer program which represents the model, that causes the confusion between model and program so well described by Evans et. al. (Ref. 3 page 5). However, the important thing is that the computer program is not the model, but rather represents the operation of the model.

The great advantage of simulation studies is that with the powerful digital computers available, it is not difficult to write programs representing highly complex operations. Non-linearities are readily handled and by using pseudo-random number generators stochastic processes can also be simulated. Therefore operations which are too difficult to examine analytically can quite easily be simulated.

Unfortunately simulation does not solve problems as such, but only shows what the model would do under certain circumstances. Therefore a simulation study consists of examining the operation of the model under a wide variety of circumstances and it is highly desirable to compare its operation with the real system. If the model and the system behave in a similar way, it is then possible to examine the operation of the model under different circumstances and use this to predict how the real system would operate under similar circumstances. In this way it is hoped to find the circumstances under which the model (and hence the real system) operates optimally (as determined by some, often arbitary, performance index or indices).

When a computer program is used to represent the model it is possible to include in the program a measure of the performance indices and hence it is not necessary to observe the actual behaviour of the model, but only the value of the performance indices.

1.3 AIM OF THIS STUDY.

The aim of this study is to write a program which can be used to simulate a distribution system so that various methods of truck dispatching can be compared. In the system considered each of the customers will receive deliveries on one or more fixed days (this is a fairly good assumption for the delivery of many consumer goods, with the provision that supplementary deliveries can and will be made if the customer runs out of stock), the quantity to be delivered depending upon the customer's stock the previous day, which in turn depends upon the consumption since the previous delivery. This consumption of course is not constant but varies according to the weather and other such external forces. Hence the quantity to be delivered will vary from week to week. Under such a system it is difficult for the truck dispatcher to optimize the routes/

routes/ followed by the trucks unless he carries out fairly complex calculations every day, which even with the use of a computer can be both time consuming and expensive. Therefore, the normal method is to have a standard set of routes for each day of the week. The problem being to choose a set of standard routes which over a period of time will prove to be optimum. An obvious method is to apply an analytical technique to optimize the routes for the average demand and to use these.

However, as pointed out above, it is also necessary to consider such things as the capital costs of both trucks and containers. It is here that a simulation of the system becomes advantageous. Delivery routes can be determined by some analytical means and these can be simulated; the routes can then be slightly altered in the light of the performances of the model and the new routes simulated. This process can be repeated and the best set of routes determined. These can then be used as the standard routes by the truck dispatcher.

A problem related to the truck dispatching problem is the selection of depot and factory sites. The simulation model described in this paper can also be used to simulate the effect of different depot and factory locations to assist in the determination of the best site.

In summary the aim is to provide a method of rapidly simulating and comparing the efficiency of several methods of dispatching trucks from a central depot or warehouse to a number of customers.

2. MODEL.

In this chapter will be described the model developed of a truck distribution system, and how this model relates to real systems and the probems incurred in truck dispatching. The next chapter will describe the computer program written to simulate the operation of this model.

2.1 DEPOT, OUTLETS AND CONTAINERS.

In the model containers are delivered from a central depot to a number of outlets. The term container is used rather than product or good as a special interest of this study is the case where the product is delivered in containers owned by the distributor. A container might be a case of soft drinks, a churn of ice-cream or a cylinder of gas; in situations where containers in this sense are not used, such as the delivery of a ready mixed concrete, the term container can refer to a unit of product such as a cubic yard or ton etc. In any case the quantity to be delivered is expressed as an integral number of containers. By keeping a record of the variation in the total number of containers in the field, it is possible to determine how many containers are needed to satisfy the demand. By repeated simulation using different delivery frequencies it is possible to obtain the number of containers required as a function of delivery frequency. In this way distribution costs can be compared with container costs. For cases such as the delivery of petrol where the size of bulk tanks of the service station depends upon the delivery frequency, a similar comparison can be made by determining the delivery frequencies required for differentsizes of tanks.

The depot in the model can represent a factory, warehouse or an actual depot; in any case it is the source of containers and the containers are returned to the depot when their contents are consumed.

In the case of the distribution of petrol the containers are not returned. However, if this is the situation this part of the model can be ignored without the results being affected.

The outlets are used to represent the point to which deliveries are made. In most cases they will be retail shops or the like, but they may also include subsidiary depots, from which further deliveries are made. For this model however, such depots are considered end points in the delivery chain. It is assumed that the contents of the containers are consumed at the outlets at a rate which follows some known statistical distribution. In the real situation the containers are possibly purchased by a customer from the outlet and consumed at his home, the empty container being returned to the outlet. However, by choosing the correct statistical distribution, the model can be made to represent this situation.

2.2 TRUCKS AND ROUTES.

For the truck dispatching problem the most important things to consider are the trucks and the routes they follow in making deliveries to outlets. The routes must be chosen so that the trucks have sufficient capacity to make the necessary deliveries to each of the outlets on the route, and also have sufficient time to complete the route in one day.

In the model, "orders" are taken each day from each of the outlets which are to receive a delivery the following day. The "order" is calculated by updating the stock of the outlet to the end of the day and comparing this with a predetermined "ideal stock". The difference between the "actual stock" and "ideal stock" is taken as the "order". When all of the "orders" for a particular route are determined the total number of containers to be delivered on that route is determined and a truck is allocated to that route.

If the truck capacity is insufficient a note is made of this fact. If there are insufficient trucks to allocate one to each route, then the first trucks to return to the depot after completing their routes are allocated to the remaining routes.

As mentioned above the routes to be followed by the trucks are determined bymethods not covered in this paper. It is assumed however, that some analytical technique would be used.

It should be noted that an outlet may be included in more than one route and that two or more routes may be identical. This is because a route represents a sequence of outlets to which deliveries are made on a particular day.

2.3 CONSUMPTION AND DELIVERIES.

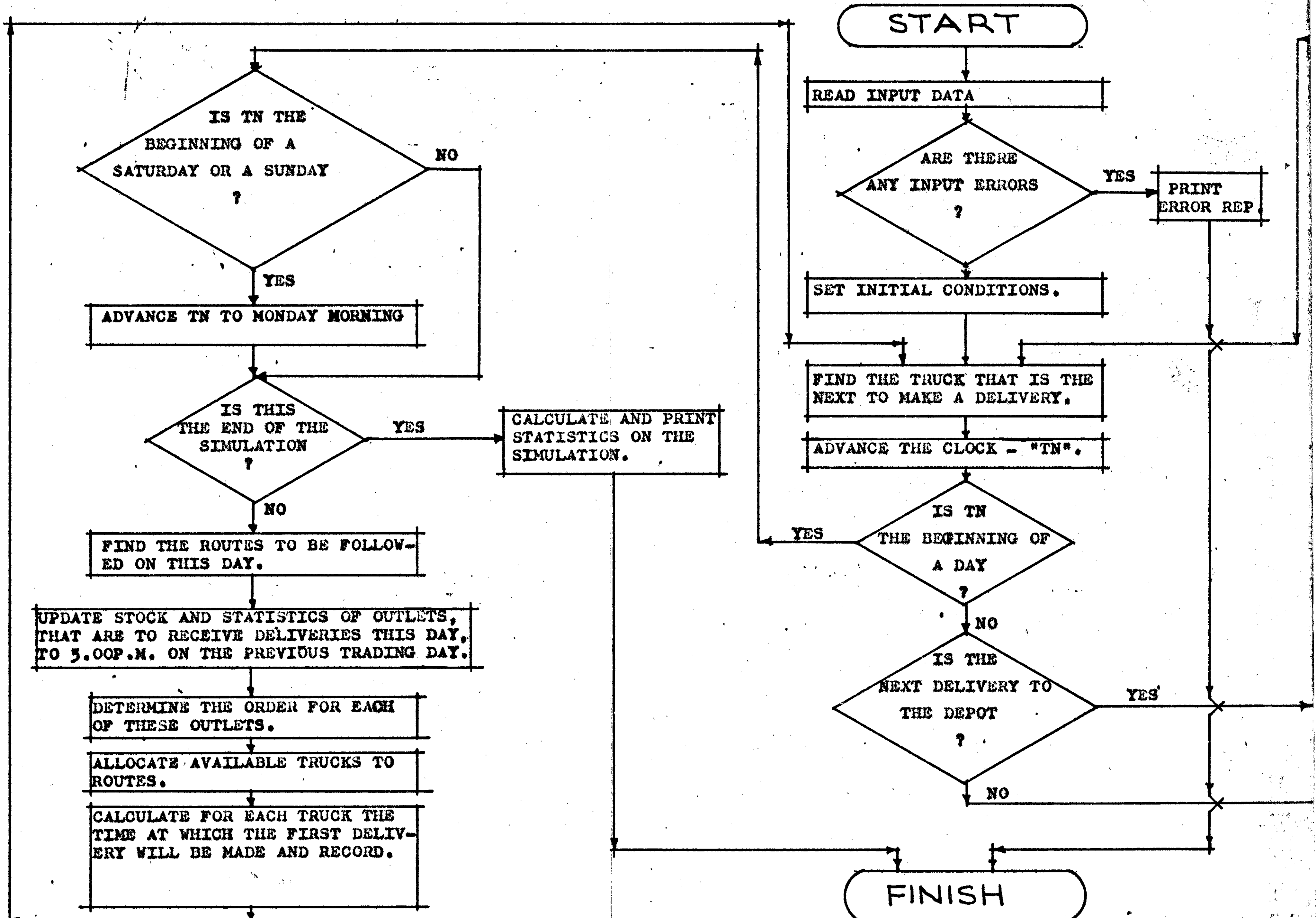
The time taken for a truck to complete a route is determined by finding the time a truck takes to drive from one outlet to another and make a delivery. The average time for this is determined using data provided by a work study and then the simulated time is generated by sampling a standard statistical distribution with mean equal to the average time for the delivery. This is a key part of the handling of the passage of time. The simulation time is advanced every time a delivery is made, a calculation is then made of the time to the next delivery and the simulation time again advanced. A complete explanation of this method of timing is contained in section 3.5.

Consumption is determined in a similar way to delivery times. An average time to consume a container is determined from the average annual consumption and then a statistical distribution having this average time as a mean is sampled to determine the actual time to consume the container.

A record is kept of when the next container will be consumed. Each time a delivery is made to an outlet, or an "order" taken, a check is made to see if a container has been consumed since the last check. If there has, the stock of the outlet is updated and the time taken for the next container to be consumed is calculated. If this container was also consumed since the last check the process is repeated until the outlet's stock is up to date at the time of the delivery or"order". A check is made to see if the stock of an outlet falls to zero between deliveries, if this occurs it is noted.

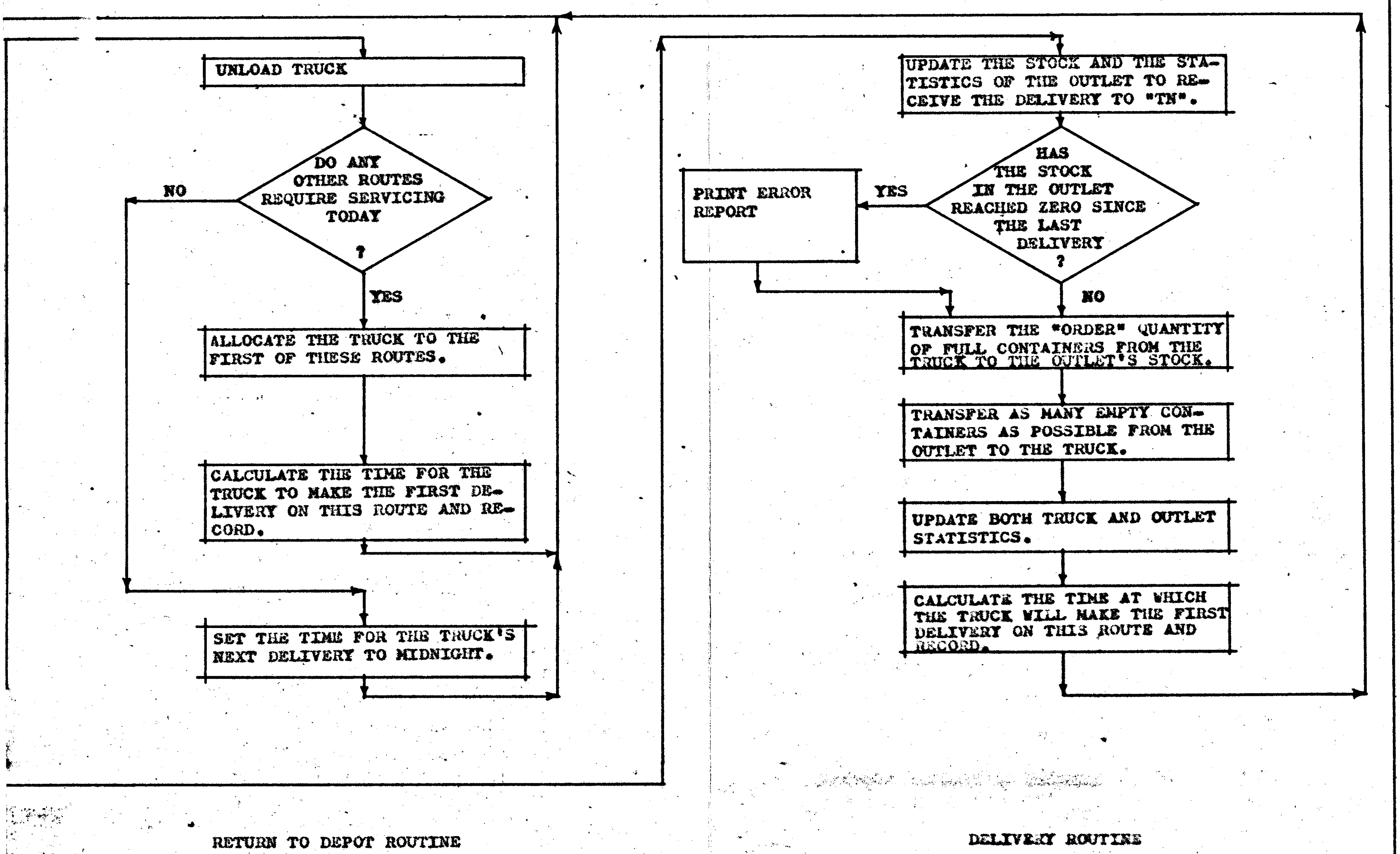
After the stock of the outlet is updated to the time of a delivery as explained in the previous paragraph, a number of containers, equal to the "order" are transferred from the truck to the outlet. Also if there is sufficient room on the truck, the empty containers in the outlet are transferred to the truck for return to the depot.

When the truck has completed a route it returns to the depot and all the empty containers on the truck are transferred to the depot stock. In this program no record is kept of the depot stock as this is considered a problem separate from that of the dispatching of trucks. It would not, however, be difficult to incorporate the depot stock in the model (or the computer program discussed in the next chapter) if it was an essential part of a particular study.



BEGINNING OF DAY ROUTINE.

MAINLINE ROUTINE.



COMPUTER FLOW-CHART

FIG. 1

LISTING OF FORTRAN PROGRAM

FIG. 2

```
C           SUBROUTINES  AND FUNCTION SUBPROGRAMS
        SUBROUTINE STATS(LWEEK,TN)
C
C THIS SUBROUTINE CALCULATES THE FINAL STATISTICS ON THE SIMULATION
C
        COMMON/STATUS/ROUTE(50,21),TRUCK(5,16),OUTLET(100,25),AMAX,CMAX,EM
       1AX
        INTEGER AMAX,EMAX,CMAX
        TWEEK(TN) =   TN/(24.0 * 7.0)
        TDAY(TN)  =    (TN - TWEEK(TN)*7.0*24.0)/24.0
        NNN=EMAX+2
  98765 FORMAT('1T H E   F O L L O W I N G   P A G E S   G I V E   F U L L
       1   D E T A I L S   O F   T H E   S I M U L A T I O N')
  98766 FORMAT('0A N D   R E L E V A N T   R E S U L T S   A N D   S T A T
       1 I S T I C S ')
  98767 FORMAT('0DETAILS')
  98768 FORMAT('              NUMBER OF WEEKS SIMULATED      =',I4)
  98769 FORMAT('              NUMBER OF OUTLETS CONSIDERED =',I4)
  98770 FORMAT('              NUMBER OF ROUTES CONSIDERED   =',I4)
  98771 FORMAT('              NUMBER OF TRUCKS CONSIDERED   =',I4)
  98772 FORMAT('ORESULTS')
  98773 FORMAT('              TOTAL NUMBER OF UNITS CONSUMED  =',F9.0)
  98774 FORMAT('              TOTAL NUMBER OF UNITS DELIVERED =',F9.0)
  98775 FORMAT('              AVERAGE FULL STOCK IN OUTLETS   =',F9.0)
  98776 FORMAT('              AVERAGE TOTAL STOCK IN OUTLETS  =',F9.0)
  98777 FORMAT('0             AVERAGE DELIVERY TIME PER ROUTE =',F7.3)
  98778 FORMAT('              AVERAGE DELIVERY TIME PER UNIT  =',F7.3)
  98779 FORMAT('0             MAXIMUM TOTAL STOCK IN OUTLETS  =',F9.0,'  TIME OF
       1MAX = WEEK #',I4,' DAY #',I2)
  98780 FORMAT('0-----------------------------------------------------------
       1---------------------------------------------------------')
  98781 FORMAT('OR O U T E   S T A T I S T I C S')
  98782 FORMAT('0           ROUTE NUMBER          AVERAGE TIME PER ROUTE')
  98783 FORMAT('0',8X,I7  ,16X,F7.3)
  98784 FORMAT('1T R U C K   S T A T I S T I C S')
  98785 FORMAT('OTRUCK #   AV. TIME/ROUTE   AV. HOURS/DAY   MAX TIME/ROUTE
       1   MAX ROUTE #   MIN TIME/ROUTE   MIN ROUTE #   ROUTE TIME S-D')
  98786 FORMAT('0', I6 ,8X,F7.3,9X,F7.3,10X,F7.3,8X,F6.0,10X,F7.3,9X,F4.0,
       18X,F9.3)
  98787 FORMAT('0ALL TRUCKS',4X,F7.1,9X,F7.3,69X,F9.3)
  98788 FORMAT('1O U T L E T   S T A T I S T I C S ')
  98789 FORMAT('0OUTLET   MAX   TIME   MIN   TIME   MAX   TIME   MIN
       1TIME    AV    AV   S-D   S-D   # OF   UNITS   UNITS')
  98790 FORMAT(' NUMBER  TOTAL MAX TOT  TOT  MIN TOT  FULL MAX FULL FULL M
       1IN FULL  TOT   FULL  TOTAL  FULL   DELIVS  DELIV CONSUM')
  98791 FORMAT('0', I5 ,F8.0,F7.0,F6.0,F7.0,F8.0,F7.0,F7.0,F7.0,F8.0,F7.0,
       1F6.1,F7.1,F7.0,F8.0,F8.0)
  98792 FORMAT('0ALL',3X,F7.0,F8.0,F6.0,F7.0,F8.0,F7.0,F7.0,F7.0,F8.0,F7.0
       1,F6.1,F7.1,F7.0,F8.0,F8.0)
        WRITE(3,98765)
        WRITE(3,98766)
        WRITE(3,98767)
        WRITE(3,98768) LWEEK
        WRITE(3,98769) EMAX
        WRITE(3,98770) AMAX
        WRITE(3,98771) CMAX
        WRITE(3,98772)
C                                CALCULATION OF RESULTS
```

```
      $CON=0
      $DEL=0
      DO 09876 I=1,EMAX
      $CON=$CON+OUTLET(I,22)
09876 $DEL=$DEL+OUTLET(I,19)
      $AVFS=OUTLET(NNN,25)/TN
      $AVTS=OUTLET(NNN,24)/TN
      RTIME=0
      RNO=0
      DO 09877 I=1,AMAX
      RTIME=RTIME+ROUTE(I,5)
09877 RNO=RNO+ROUTE(I,4)
      AVRT=RTIME/RNO
      AVUT=RTIME/$DEL
      WRITE(3,98773) $CON
      WRITE(3,98774) $DEL
      WRITE(3,98775) $AVFS
      WRITE(3,98776) $AVTS
      WRITE(3,98777) AVRT
      WRITE(3,98778) AVUT
      NTW=TWEEK(OUTLET(NNN,9))
      NTD=TDAY(OUTLET(NNN,9))
      WRITE(3,98779) OUTLET(NNN,8),NTW,NTD
C           OUTPUT OF ROUTE STATISTICS
      WRITE(3,98780)
      WRITE(3,98781)
      WRITE(3,98782)
      DO 09878 I=1,AMAX
      RAV=ROUTE(I,5)/ROUTE(I,4)
      WRITE(3,98783) I,RAV
09878 CONTINUE
C
C           OUTPUT OF TRUCK STATISTICS
      WRITE(3,98784)
      WRITE(3,98785)
      TT8 =0
      TT13=0
      TT14=0
      DO 09879 I=1,CMAX
      TT8=TT8+TRUCK(I,8)
      TT13=TT13+TRUCK(I,13)
      TT14=TT14+TRUCK(I,14)
      T2=TRUCK(I,8)/TRUCK(I,13)
      T3=TRUCK(I,8)*24.0/TN
      T8=SQRT((TRUCK(I,14)/TRUCK(I,13))-(TRUCK(I,8)/TRUCK(I,13))**2)
      WRITE(3,98786) I,T2,T3,TRUCK(I,9),TRUCK(I,10),TRUCK(I,11),TRUCK(I,
     112),T8
09879 CONTINUE
      TT2=TT8/TT13
      TT3=TT8*24.0/(TN*CMAX)
      TT8T=SQRT((TT14/TT13)-(TT8/TT13)**2)
      WRITE(3,98787) TT2,TT3,TT8T
C         OUTPUT OF OUTLET STATISTICS
      WRITE(3,98788)
      WRITE(3,98789)
      WRITE(3,98790)
      OT12=0
      OT17=0
```

```
      OT18=0
      OT19=0
      OT22=0
      OT24=0
      OT25=0
      DO 09880 I=1,EMAX
      OT12=OT12+OUTLET(I,12)
      OT17=OT17+OUTLET(I,17)
      OT18=OT18+OUTLET(I,18)
      OT19=OT19+OUTLET(I,19)
      OT22=OT22+OUTLET(I,22)
      OA10=OUTLET(I,24)/TN
      OA11=OUTLET(I,25)/TN
      OA12=SQRT(OUTLET(I,12)/TN-(OUTLET(I,24)/TN)**2)
      OA13=SQRT(OUTLET(I,17)/TN-(OUTLET(I,25)/TN)**2)
      WRITE(3,98791) I,OUTLET(I,8),OUTLET(I,9),OUTLET(I,10),OUTLET(I,11)
     1,OUTLET(I,13),OUTLET(I,14),OUTLET(I,15),OUTLET(I,16),OA10,OA11,OA1
     12,OA13,OUTLET(I,18),OUTLET(I,19),OUTLET(I,22)
09880 CONTINUE
C                      CALCULATION OF OVERALL OUTPUT STATISTICS
      OTT10=OUTLET(NNN,24)/TN
      OTT11=OUTLET(NNN,25)/TN
      OTT12=SQRT((OUTLET(NNN,12)/TN)-OTT10**2)
      OTT13=SQRT((OUTLET(NNN,17)/TN)-OTT11**2)
      WRITE(3,98792) OUTLET(NNN,8),OUTLET(NNN,9),OUTLET(NNN,10),OUTLET(N
     1NN,11),OUTLET(NNN,13),OUTLET(NNN,14),OUTLET(NNN,15),OUTLET(NNN,16)
     2,OTT10,OTT11,OTT12,OTT13,OT18,OT19,OT22
      RETURN
      END
```

```
      SUBROUTINE TOTAL(TN)
C
C THIS SUBROUTINE CALCULATES THE TOTAL # OF CONTAINERS IN THE OUTLETS
C
      COMMON/STATUS/ROUTE(50,21),TRUCK(5,16),OUTLET(100,25),AMAX,CMAX,EM
     1AX
      INTEGER AMAX,EMAX,CMAX
      $FULL=0
      $MT=0
      NNN=EMAX+2
      DO 12121 I=1,EMAX
      $FULL=$FULL+OUTLET(I,5)
      $MT=$MT+OUTLET(I,6)
12121 CONTINUE
      $TOT=$FULL+$MT
      IF($TOT.LT.OUTLET(NNN,8)) GO TO 12122
      OUTLET(NNN,8)=$TOT
      OUTLET(NNN,9)=TN
12122 IF($TOT.GT.OUTLET(NNN,10)) GO TO 12123
      OUTLET(NNN,10)=$TOT
      OUTLET(NNN,11)=TN
12123 IF($FULL.LT.OUTLET(NNN,13)) GO TO 12124
      OUTLET(NNN,13)=$FULL
      OUTLET(NNN,14)=TN
12124 IF($FULL.GT.OUTLET(NNN,15)) GO TO 12125
      OUTLET(NNN,15)=$FULL
      OUTLET(NNN,16)=TN
12125 OUTLET(NNN,12)=OUTLET(NNN,12)+(TN-OUTLET(NNN,23))*($TOT**2)
      OUTLET(NNN,17)=OUTLET(NNN,17)+(TN-OUTLET(NNN,23))*($FULL**2)
      OUTLET(NNN,24)=OUTLET(NNN,24)+(TN-OUTLET(NNN,23))*$TOT
      OUTLET(NNN,25)=OUTLET(NNN,25)+(TN-OUTLET(NNN,23))*$FULL
      OUTLET(NNN,23)=TN
      RETURN
      END
```

```
      SUBROUTINE ERROR1 (CARD,CARDI,TYPE,ERROR)
C
C THIS SUBROUTINE IS USED TO PRINT OUT ANY ERRORS OCURRING DURING INPUT
C
      INTEGER CARD,TYPE,ERROR,CARD1
66800 FORMAT('0 AN ERROR HAS OCURRED IN THE',I3,'CARD READ OF TYPE',I2,
     1'. THE INCORRECT DATA GIVEN IS',F5.2,'. THIS IS ERROR # ', I2)
      ERROR = ERROR + 1
      WRITE(3,66800) CARD,TYPE,CARDI,ERROR
      RETURN
      END
```

```
      SUBROUTINE RANDU(IX,IY,YFL)
C
C THIS SUBROUTINE CALCULATES A RANDOM NUMBER YFL
C
      IY = IX * 65539
      IF(IY) 5,6,6
    5   IY = IY + 2147483647 +1
    6   YFL = IY
      YFL = YFL * .4656613E-9
      RETURN
      END
```

```
      SUBROUTINE OUTPUT
C
C THIS SUBROUTINE PRINTS OUT THE MATRICES ROUTE,TRUCK &OUTLET
C
      COMMON/STATUS/ROUTE(50,21),TRUCK(5,16),OUTLET(100,25),AMAX,CMAX,EM
     1AX
      INTEGER AMAX,EMAX,CMAX
55560 FORMAT('1                                                   F I N A L   S T
     1A T U S   O F   R O U T E S  ')
55561 FORMAT('0 ROUTE    TRUCK     TIME    NO. OF  TOTAL    NO. OF  COL.
     1NO. OF   DAY      CODE  NUMBERS  OF  OUTLETS  ON  ROUTE ')
55562 FORMAT('  NUMBER   NUMBER   STARTED  RUNS    TIME  OUTLETS    NO.
     1 CONT.  SERVED   1   2    3    4    5    6    7    8    9  10  11  12 ')
55563 FORMAT('0',F6.0,F8.0,F9.1,F7.0,F9.1,F7.0,F8.0,F8.0,F8.0,F6.0,11F4.
     10 )
55564 FORMAT('1                                                F I N A L   S T A T
     1 U S   O F   T R U C K S  ')
55565 FORMAT('0 TRUCK     CALL    ROUTE     NEXT    NO. OF  NO. OF   TIME
     1 TOTAL    MAX     NUMBER   MIN     NUMBER  NO. OF  SUM OF    TRUCK ')
55566 FORMAT('  NUMBER    TIME    NUMBER  OUTLET   FULL    EMPTY   STARTED
     1 TIME     TIME    OF MAX   TIME    OF MIN  ROUTES SQUARES CAPACITY ')
55567 FORMAT('0',F5.0,F10.1,F6.0,F9.0,F7.0,F8.0,F10.1,F8.1,F8.1,F6.0,F10
     1.1,F6.0,F9.0,F10.1,F5.0    )
55568 FORMAT('1                                                F I N A L   S T A T
     1 U S   O F   O U T L E T S ' )
55569 FORMAT('0 OUTLET      X        Y     ANNUAL  NO. OF  NO. OF    NEXT
     1 MAX      TIME     MIN     TIME    SUM OF SQUARES')
55570 FORMAT('  NUMBER  CO-ORD  CO-ORD  CONSUM   FULL   EMPTY     TIME
     1 TOTAL    MAXT     TOTAL    MINT    FOR TOTAL STOCK ')
55571 FORMAT('0',F6.0,F9.1,F8.1,F7.0,F8.0,F8.0,F9.1,F7.0,F9.1,F7.0,F9.1,
     1F13.1 )
55572 FORMAT('OCONTINUATION')
55573 FORMAT('0 OUTLET    MAX       TIME     MIN      TIME    SUM OF  NO. OF
     1CONT.     NEXT    IDEAL   CONT.   TIME OF LAST   AVERAGE  AVERAGE')
55574 FORMAT('  NUMBER    FULL    MAXFULL  FULL    MINFULL SQUARE   DELIVS
     1DELIV.  DELIV.   STOCK  CONSUM.   DELIVERY       TOTAL     FULL')
55575 FORMAT('0',I6  ,F8.0,F9.1,F6.0,F10.1,F9.1,F6.0,F8.0,F7.0,F9.0,F8.0
     1,F11.1,5X,F9.1,F8.1    )
      WRITE(3,55560)
      WRITE(3,55561)
      WRITE(3,55562)
      WRITE(3,55563)  ((ROUTE(I,J),J=1,21),I=1,AMAX)
      WRITE(3,55564)
      WRITE(3,55565)
      WRITE(3,55566)
      WRITE(3,55567)  ((TRUCK(I,J),J=1,15),I=1,CMAX)
      WRITE(3,55568)
      WRITE(3,55569)
      WRITE(3,55570)
      WRITE(3,55571)  ((OUTLET(I,J),J=1,12),I=1,EMAX)
      WRITE(3,55572)
      WRITE(3,55573)
      WRITE(3,55574)
      DO 55576 I=1,EMAX
      WRITE(3,55575) I,(OUTLET(I, J),J=13,25)
55576 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INPUT
C
C THIS SUBROUTINE READS DATA ON STATISTICAL DISTRIBUTIONS OF TIMES
C
      COMMON/DIST/DIST1,DIST2,DIST3,DIST4,DIST5
      INTEGER Z8,Z9,Z10,X8,X9,X10
77777 FORMAT(F5.2,I5)
77780 FORMAT('1THERE IS AN ERROR IN CARD TYPE ',I5)
      READ(1,77777) DIST1,X8
      Z8=8
      IF(X8.NE.8) WRITE(3,77780) Z8
      READ(1,77777) DIST2,X9
      Z9=9
      IF(X9.NE.9) WRITE(3,77780) Z9
77778 FORMAT(3F5.2,I5)
      READ(1,77778) DIST3,DIST4,DIST5,X10
      Z10=10
      IF(X10.NE.10) WRITE(3,77780) Z10
      RETURN
      END
```

```
      SUBROUTINE SERTT(YFL,STT,A,B,C)
C
C THIS SUBROUTINE CALCULATES THE TRUCKS TIME TO MAKE THE NEXT DELIVERY
C
      COMMON/STATUS/ROUTE(50,21),TRUCK(5,16),OUTLET(100,25),AMAX,CMAX,EM
     1AX
      COMMON/DIST/DIST1,DIST2,DIST3,DIST4,DIST5
      ALTH=SQRT((OUTLET(A,2)-OUTLET(B,2))**2+(OUTLET(A,3)-OUTLET(B,3))**
     12)
      TRT=ALTH*1.2*DIST3
      CT=C*DIST4
      STT=(100-DIST2+2*DIST2*YFL)*(TRT+CT+DIST5)/6000.0
      RETURN
      END
```

```
      SUBROUTINE SERTO(YFL,ANCONS,TI,STO,MINSH,MAXSH)
C
C THIS SUBROUTINE CALCULATES THE TIME TO CONSUME A CONTAINER
C
      IWEEK(B)=B/(24*7)
      IDAY(B)=(B-IWEEK(B)*7*24)/24
      HHOUR(B)=B-(IWEEK(B)*7+IDAY(B))*24
      COMMON/DIST/DIST1,DIST2,DIST3,DIST4,DIST5
      STO=(100-DIST1+2*DIST1*YFL)*312*(MAXSH-MINSH)/(ANCONS*100)
      A=HHOUR(TI)+STO
      IF(A.LE.MAXSH) GO TO 77781
      ST2=STO
77782 ST2=ST2+(24-MAXSH+MINSH)
      C=ST2+HHOUR(TI)
      D=HHOUR(C)
      IF(D.GE.MAXSH) GO TO 77782
      E=IDAY(C)
      IF(E.EQ.0) C=C+24
      IF(E.EQ.6) C=C+48
      STO=C-HHOUR(TI)
77781 CONTINUE
      RETURN
      END
```

```
      BLOCK DATA
      COMMON/STATUS/ROUTE(50,21),TRUCK(5,16),OUTLET(100,25),AMAX,CMAX,EM
     1AX
      DATA ROUTE/1050*0.0/,TRUCK/80*0.0/,OUTLET/2500*0.0/
      END
```

```
      COMMON/STATUS/ROUTE(50,21),TRUCK(5,16),OUTLET(100,25),AMAX,CMAX,EM
     1AX
      COMMON/DIST/DIST1,DIST2,DIST3,DIST4,DIST5
      INTEGER THISO,THISR,THIST,TDAY,TWEEK,THOUR,AMAX,PNEXTO,CMAX,EMAX,A
     1,B,C,D,E,F,CARD,CARD1,CARD2,CARD3,CARD4,CARD5,CARD6,CARD7,TYPE1,TY
     2PE2,TYPE3,TYPE4,TYPE5,TYPE6,TYPE7,ERROR,SDAYS,DDAYS,TEST1,TEST2,TE
     3ST3
      REAL MINSH,MAXSH,MINDH,MAXDH
      TWEEK(TN) =  TN/(24.0 * 7.0)
      TDAY(TN)  =   (TN - TWEEK(TN)*7.0*24.0)/24.0
      THOUR(TN) =   TN - (TWEEK(TN)*7.0 + TDAY(TN)) *24.0
      NAMELIST/NAM1/TRUCK,ROUTE,OUTLET,TN,THIST,THISO,THISR/NAM2/TN,THI
     1ST,THISO,THISR/NAM3/AMAX,CMAX,EMAX,LWEEK,MINSH,MAXSH,SDAYS,MINDH,
     2MAXDH,DDAYS
11122 FORMAT('  AN ERROR HAS OCURRED AT 11111 DUE TO TRUCK(THIST,2) ')
11130 FORMAT('  THE FOLLOWING PAGES GIVE THE INITIAL STATUS')
11131 FORMAT('  OUTLET #',I5,' RAN OUT OF STOCK AT TIME',F6.1, '
     1THE STOCK WAS REPLENISHED AT TIME ',F6.1)
11132 FORMAT('0  TRUCK# ',F4.0,'IS TO SMALL TO TAKE LOAD OF ',F5.0,
     1'CONTAINERS FOR ROUTE #',I4,'ONDAY',I1,'OF WEEK ',I4)
11133 FORMAT('1 THE FOLLOWING PAGES GIVE THE STATUS AT THE BEGINNING OF
     1DAY # ',I6,' OF WEEK #  ',I6 )
11134 FORMAT(' TRUCK # ',I3,' HAS A LOAD OF ',F5.0,' CONTAINERS WHICH IS
     1LESS THAN ORDER OF ',F5.0,'CONTAINERS FOR OUTLET #',I4,'ON DAY #'
     2,I1,'OF WEEK  #  ',I5)
11135 FORMAT(' TRUCK # ',I3,'HAS RUN OUT OF STOCK ON ROUTE #',I3,'DUR
     1ING WEEH # ',I5,'AFTER OUTLET #',I5)
C
C THIS SECTION READS THE INPUT DATA AND SETS INITIAL CONDITIONS
C
      THIST = 1
      TN = 0.0
      IX =  55557
      TYPE1 = 1
      TYPE2 = 2
      TYPE3 = 3
      TYPE4 = 4
      TYPE5 = 5
      TYPE6 = 6
      TYPE7 = 7
      ERROR = 0
66666 FORMAT(16I5)
66667 FORMAT(I5,2F5.2,I5,2F5.2,2I5)
66668 FORMAT(I5,2F5.2,5I5)
66669 FORMAT('1 I N P U T  E R R O R  R E P O R T ' )
66670 FORMAT('0 T H E R E  H A S  B E E N ',I2,' E R R O R S  I N  T H E
     1  I N P U T ')
66671 FORMAT('0',10F8.2)
      WRITE(3,66669)
      READ(1,66666) AMAX,CMAX,EMAX,CARD1
      CARD = 1
      IF(CARD1.NE.1) CALL ERROR1(CARD,CARD1,TYPE1,ERROR)
      READ(1,66667) LWEEK,MINSH,MAXSH,SDAYS,MINDH,MAXDH,DDAYS,CARD2
      CARD = CARD + 1
      WRITE(3,NAM3)
      IF(CARD2.NE.2) CALL ERROR1(CARD,CARD2,TYPE1,ERROR)
      IF(AMAX.GT.50) CALL ERROR1(CARD,AMAX,TYPE2,ERROR)
      IF(CMAX.GT.5)  CALL ERROR1(CARD,CMAX,TYPE3,ERROR)
```

```
      IF(EMAX.GT.98) CALL ERROR1(CARD,EMAX,TYPE4,ERROR)
      NNN=EMAX+2
      OUTLET(NNN,10)=100000
      OUTLET(NNN,15)=100000
      DEPOT = 1+EMAX
      DO 66700 I = 1,AMAX
      READ(1,66666) TEST1,IR6,IR9,CARD3
      ROUTE(I,6) = IR6
      ROUTE(I,9) = IR9
      CARD = CARD + 1
      IF(CARD3.NE.3) CALL ERROR1(CARD,CARD3,TYPE1,ERROR)
      ROUTE(I,1) = I
      IF(TEST1.NE.I) CALL ERROR1(CARD,TEST1,TYPE5,ERROR)
      READ(1,66666) TEST1,IR10,IR11,IR12,IR13,IR14,IR15,IR16,IR17,IR18,I
     1R19,IR20,IR21,CARD4
      ROUTE(I,10) = IR10
      ROUTE(I,11) = IR11
      ROUTE(I,12) = IR12
      ROUTE(I,13) = IR13
      ROUTE(I,14) = IR14
      ROUTE(I,15) = IR15
      ROUTE(I,16) = IR16
      ROUTE(I,17) = IR17
      ROUTE(I,18) = IR18
      ROUTE(I,19) = IR19
      ROUTE(I,20) = IR20
      ROUTE(I,21) = IR21
      CARD = CARD + 1
      WRITE(3,66671) (ROUTE(I,J),J=1,21)
      IF(CARD4.NE.4) CALL ERROR1(CARD,CARD4,TYPE1,ERROR)
      IF(TEST1.NE.I) CALL ERROR1(CARD,TEST1,TYPE5,ERROR)
      IF(ROUTE(I,10).LE.0) CALL ERROR1(CARD,ROUTE(I,10),TYPE6,ERROR)
      N11 = 10 + ROUTE(I,6)
      IF(ROUTE(I,N11).NE.0 ) CALL ERROR1(CARD,ROUTE(I,N11),TYPE7,ERROR)
      ROUTE(I,6)=ROUTE(I,6)+1
      ROUTE(I,(9+ROUTE(I,6)))=DEPOT
66700 CONTINUE
      DO 66701 I = 1,CMAX
      READ(1,66666) TEST2,IT15,CARD5
      TRUCK(I,15) = IT15
      CARD = CARD + 1
      IF(CARD5.NE.5) CALL ERROR1(CARD,CARD5,TYPE1,ERROR)
      TRUCK(I,1) = I
      IF(TEST2.NE.I) CALL ERROR1(CARD,TEST2,TYPE5,ERROR)
      TRUCK(I,11)=100000
66701 CONTINUE
      DO 66702 I =1,EMAX
      OUTLET(I,1) = I
      READ(1,66668) TEST3,OUTLET(I,2),OUTLET(I,3),I04,I05,I06,I021,CARD6
      OUTLET(I,4) = I04
      OUTLET(I,5) = I05
      OUTLET(I,6) = I06
      OUTLET(I,21) = I021
      CARD = CARD + 1
      IF(CARD6.NE.6) CALL ERROR1(CARD,CARD6,TYPE1,ERROR)
      IF(TEST3.NE.I) CALL ERROR1(CARD,TEST3,TYPE5,ERROR)
      OUTLET(I,8)=I06          +I05
      OUTLET(I,13)=I05
```

```
      OUTLET(I,10)=I05+I06
      OUTLET(I,15)=I05
66702 CONTINUE
      READ(1,66666) ID2,ID3,CARD7
      OUTLET(DEPOT,2) = ID2
      OUTLET(DEPOT,3) = ID3
      CARD = CARD + 1
      IF(CARD7.NE.7) CALL ERROR1(CARD,CARD7,TYPE1,ERROR)
      OUTLET(DEPOT,1)=DEPOT
      CALL INPUT
      WRITE(3,66670) ERROR
      WRITE(3,11130)
      CALL TOTAL(TN)
      CALL OUTPUT
C
C T H I S   I S   T H E   M A I N L I N E   R O U T I N E
C
11111 DO 11110  C = 1, AMAX
      IF (TRUCK(C,2).LT.TRUCK(THIST,2))     THIST = C
11110 CONTINUE
      IF(TRUCK(THIST,2).LT.TN) GO TO 11121
      TN =   TRUCK(THIST,2)
C       CHECK FOR BEGINNING OF DAY
      IF (THOUR(TN).EQ.0)    GO TO 22222
      THISO = TRUCK (THIST,4)
      IF(THISO.EQ.DEPOT)  GO TO 33333
C
C D E L I V E R Y   R O U T I N E   P A R T   1
C
      IF(TN.LT.OUTLET(THISO,7))  GO TO 44444
C       ADJUST  OUTLET  STOCK
11117 OUTLET(THISO,5) = OUTLET(THISO,5) - 1
      OUTLET(THISO,6) = OUTLET(THISO,6) + 1
      OUTLET(THISO,22) = OUTLET(THISO,22) + 1
      IF(OUTLET(THISO,5).EQ.0 )    GO TO 11118
      IF(OUTLET(THISO,5).GE.OUTLET(THISO,15)) GO TO 11119
      OUTLET(THISO,15) = OUTLET(THISO,5)
      OUTLET(THISO,16) = OUTLET(THISO,7)
C       CALCULATION  OF  SERVICE TIME FOR OUTLET ( STO )
11119   IX = IY
      CALL  RANDU (IX,IY,YFL)
      CALL  SERTO(YFL,OUTLET(THISO,4),OUTLET(THISO,7),STO,MINSH,MAXSH)
      OUTLET(THISO,7) = OUTLET(THISO,7 )  + STO
      IF (OUTLET(THISO,7).LE.TN)  GO TO 11120
C       CALCULATION OF STANDARD DEVIATION OF FULL CONTAINERS
      OUTLET(THISO,17)=OUTLET(THISO,17)+STO*((OUTLET(THISO,5))**2.0)
      OUTLET(THISO,25)=OUTLET(THISO,25)+STO*OUTLET(THISO,5)
      GO  TO  44444
11120 OUTLET(THISO,17) = OUTLET(THISO,17) + STO*((OUTLET(THISO,5))**2.0)
      OUTLET(THISO,25)=OUTLET(THISO,25)+STO*OUTLET(THISO,5)
      GO TO 11117
C       OUT OF  STOCK REPORT  ( TEMPORARY )
11118 CONTINUE
      WRITE(3,11131) THISO,OUTLET(THISO,7),TN
      GO TO 44444
11121 WRITE(3,11122)
      WRITE(3,NAM1)
      GO TO 55555
```

```
C
C BEINNING  OF  DAY  ROUTINE
C
22222 CONTINUE
      IF (TDAY(TN).EQ.0) TN=TN+24.0
      IF (TDAY(TN).EQ.6)    TN = TN + 48.0
C        CHECK FOR END OF SIMULATION
      IF ( TWEEK(TN).EQ.LWEEK)   GO  TO  55555
C       I IS ROUTE NO. AND  T IS  TRUCK NO.
      I = 0
      T = 0
      TNO  = TN - 7.0
22223   I = I + 1
      IF(I.GT.AMAX)  GO TO 22230
      ROUTE(I,8)  =  0
      IF (TDAY(TN).NE.ROUTE(I,9))  GO TO  22223
C       N IS OUTLET  NO.
      N= 0
22224 N= N + 1
      IF (N.GT.ROUTE(I,6))  GO TO 22225
C       M IS COLUMN NO.  OF OUTLET CODE NO.
      M =  N + 9
      OUT =  ROUTE(I,M)
      IF(OUT.EQ.DEPOT) GO TO 22224
C
C      N O T E
C
C     THE CALCULATION BELOW ADJUSTS THE OUTLETS STOCK AND STATISTICS TO
C     5.00 PM THE PREVIOUS DAY , WHERE THIS ORDERING TIME IS CALLED TNO
C
22241 IF(OUTLET(OUT,7).GT.TNO)  GO TO 22240
      OUTLET(OUT,5) = OUTLET(OUT,5) - 1
      OUTLET(OUT,6) = OUTLET(OUT,6) + 1
      OUTLET(OUT,22) = OUTLET(OUT,22) + 1
C
C     CALCULATION OF SERVICE TIME
      IX = IY
      CALL RANDU(IX,IY,YFL)
      CALL SERTO(YFL,OUTLET(OUT,4),OUTLET(OUT,7),STO,MINSH,MAXSH)
      OUTLET(OUT,7) = OUTLET(OUT,7) + STO
      OUTLET(OUT,17)=OUTLET(OUT,17)+STO*((OUTLET(OUT,5))**2)
      OUTLET(THISO,25)=OUTLET(THISO,25)+STO*OUTLET(THISO,5)
      GO TO 22241
22240 OUTLET(OUT,20) = OUTLET(OUT,21) - OUTLET(OUT,5)
      ROUTE(I,8)  = ROUTE(I,8) + OUTLET(OUT,20)
      GO  TO  22224
22230 DO 22231 C=1,AMAX
      IF(TRUCK(C,2).LE.TN) TRUCK(C,2)=TRUCK(C,2)+24.0
22231 CONTINUE
      GO TO 11111
C       SELECTION OF TRUCK
22225 T=T+1
      IF (T.LE.CMAX)  GO TO 22226
      ROUTE(I,2) = 0
      GO TO 22223
22226 IF(TRUCK(T,15).GE.ROUTE(I,8))  GO TO 22229
C       TOO SMALL A TRUCK  ROUTINE  ( TEMPORARY)
22227 CONTINUE
```

```
      ATDAY=TDAY(TN)
      AWEEK=TWEEK(TN)
      WRITE(3,11132) T,ROUTE(I,8),I,ATDAY,AWEEK
      GO TO 22229
22229  TRUCK(T,5) = ROUTE(I,8)
       TRUCK(T,6) = 0
       TRUCK(T,4) = ROUTE(I,10)
      IX = IY
      CALL RANDU(IX,IY,YFL)
      IA = TRUCK(T,4)
      CALL  SERTT(YFL,STT,IA,DEPOT,OUTLET(IA,20))
      TRUCK(T,2) = TN +MINSH + STT
      ROUTE(I,2) = T
      TRUCK(T,3) = I
      ROUTE(I,7) = 10
      TRUCK(T,7) = TN + MINSH
      ROUTE(I,3) = TN + MINSH
      ROUTE(I,4) = ROUTE(I,4) +1
      NTDAY=TDAY(TN)
      NWEEK=TWEEK(TN)
      CALL TOTAL(TN)
      GO TO 22223
C
C R E T U R N   T O   D E P O T   R O U T I N E
C
C       C A L C U L A T I O N   O F   E N D   O F   R O U T E   S T A T S.
33333 ROUTE(TRUCK(THIST,3),5)=ROUTE(TRUCK(THIST,3),5)+TN-TRUCK(THIST,7)
      TRUCK(THIST,6) = 0
      TRUCK(THIST,5) = 0.0
      TRUCK(THIST,8) = TRUCK(THIST,8) + TN - TRUCK(THIST,7)
      IF((TN-TRUCK(THIST,7)).LE.TRUCK(THIST,9)) GO TO 33336
      TRUCK(THIST,9) =  TN - TRUCK(THIST,7)
      TRUCK(THIST,10) = TRUCK(THIST,3)
33336 IF((TN-TRUCK(THIST,7)).GE.TRUCK(THIST,11)) GO TO 33337
      TRUCK(THIST,11) = TN - TRUCK(THIST,7)
      TRUCK(THIST,12) = TRUCK(THIST,3)
33337 TRUCK(THIST,13) = TRUCK(THIST,13) + 1
       TRUCK(THIST,14)=TRUCK(THIST,14)+(TN-TRUCK(THIST,7))**2
        I=0
33334  I = I + 1
       IF(I.LE.AMAX)   GO TO 33335
      TRUCK(THIST,2)  = (TWEEK(TN)*7+TDAY(TN))*24.0 +24.0
      GO   TO   11111
33335  IF(ROUTE(I,9).NE.TDAY(TN)) GO TO 33334
       IF(ROUTE(I,2).NE.0)        GO TO 33334
       TRUCK(THIST,5)   = ROUTE(I,8)
       ROUTE(I,2) = THIST
       ROUTE(I,3) = TN
       ROUTE(I,7) = 10
       TRUCK(THIST,3)   = I
       TRUCK(THIST,4)   = ROUTE(I,10)
       TRUCK(THIST,7)   =  TN
C        TRUCK  SERVICE TIME
      IX  =  IY
      CALL RANDU(IX,IY,YFL)
      IA  = TRUCK(THIST,4)
      CALL SERTT(YFL,STT,IA,DEPOT,OUTLET(IA,20))
      TRUCK(THIST,2) =  TN + STT
```

```
      GO TO 11111
C
C D E L I V E R Y   R O U T I N E   P A R T  2
C
C        S-D OF TOTAL STOCK
44444 OUTLET(THISO,12) = OUTLET(THISO,12) +(TN-OUTLET(THISO,23))*((OUTLE
     1T(THISO,5)+OUTLET(THISO,6))**2)
      OUTLET(THISO,17)=OUTLET(THISO,17)-(OUTLET(THISO,7)-TN)*(OUTLET(THI
     1SO,5)**2)
      OUTLET(THISO,24)=OUTLET(THISO,24)+(TN-OUTLET(THISO,23))*(OUTLET(TH
     1ISO,5)+OUTLET(THISO,6))
      OUTLET(THISO,25)=OUTLET(THISO,25)-(OUTLET(THISO,7)-TN)*OUTLET(THIS
     1,05)
      IF(TRUCK(THIST,5).GE.OUTLET(THISO,20))  GO TO  44445
C        PRINT ERROR ON  TRUCK  LOADING
      ATDAY=TDAY(TN)
      AWEEK=TWEEK(TN)
      WRITE(3,11134) THIST,TRUCK(THIST,5),OUTLET(THISO,20),THISO,ATDAY,
     1AWEEK
      OUTLET(THISO,20) = TRUCK(THIST,5)
C        TRANSFER  OF  FULLS
44445 OUTLET(THISO,5) = OUTLET(THISO,5) + OUTLET(THISO,20)
      OUTLET(THISO,19) = OUTLET(THISO,19) + OUTLET(THISO,20)
      TRUCK(THIST,5)  = TRUCK(THIST,5)  - OUTLET(THISO,20)
C        S-D  OF FULLS
      OUTLET(THISO,17) = OUTLET(THISO,17) + (OUTLET(THISO,7)-TN) *
     1  (OUTLET(THISO,5)**2)
      OUTLET(THISO,25)=OUTLET(THISO,25)+(OUTLET(THISO,7)-TN)*OUTLET(THIS
     10,5)
C       TRANSFER OF EMPTIES
      IF(TRUCK(THIST,15).GE.(TRUCK(THIST,5)+TRUCK(THIST,6)+OUTLET(THIST,
     16))) GO TO 44448
      LOAD = TRUCK(THIST,15)-TRUCK(THIST,5) - TRUCK(THIST,6)
      TRUCK(THIST,6) = TRUCK(THIST,6)  +LOAD
      OUTLET(THISO,6)  = OUTLET(THISO,6)  - LOAD
      GO  TO   44447
44448 TRUCK(THIST,6) = TRUCK(THIST,6) + OUTLET(THISO,6)
      OUTLET(THISO,6) = 0
44447  IF(OUTLET(THISO,5).LE.OUTLET(THISO,13)) GO TO 44449
      OUTLET(THISO,13) = OUTLET(THISO,5)
      OUTLET(THISO,14) =  TN
44449 IF(OUTLET(THISO,5)+OUTLET(THISO,6).LE.OUTLET(THISO,8)) GO TO 44460
      OUTLET(THISO,8) = OUTLET(THISO,5) + OUTLET(THISO,6)
      OUTLET(THISO,9) = TN
44460 IF(OUTLET(THISO,5)+OUTLET(THISO,6).GE.OUTLET(THISO,10)) GO TO 4446
     11
      OUTLET(THISO,10) = OUTLET(THISO,5) + OUTLET(THISO,6)
      OUTLET(THISO,11) = TN
44461 OUTLET(THISO,18) = OUTLET(THISO,18) + 1
      OUTLET(THISO,19) = OUTLET(THISO,19) + OUTLET(THISO,20) <---- RE
      OUTLET(THISO,23) = TN
C      ADJUSTMENT OF  TRUCK  RECORD
      THISR = TRUCK(THIST,3)
      ROUTE(THISR,7)  = ROUTE(THISR,7) +1
      PNEXTO = ROUTE(THISR,7)
      NEXTO =  ROUTE(THISR,PNEXTO)
       IF(NEXTO.EQ.DEPOT)  GO  TO  44450
       IF(TRUCK(THIST,5).EQ.0)  GO TO 44451
```

```
C        CALCULATION OF TRUCK SERVICE TIME
          IX = IY
          CALL  RANDU(IX,IY,YFL)
         CALL SERTT(YFL,STT,THISO,NEXTO,OUTLET(NEXTO,20))
          TRUCK(THIST,2) = TN + STT
          TRUCK(THIST,4) = NEXTO
          GO TO 11111
C          TRUCK  OUT OF STOCK BEFORE ROUTE COMPLETE  - ERROR SIGNAL
44451 CONTINUE
      AWEEK=TWEEK(TN)
      WRITE(3,11135) THIST,THISR,AWEEK,THISO
      GO TO 44450
C         RETURN OF TRUCK TO DEPOT
44450     IX = IY
       CALL RANDU(IX,IY,YFL)
      CALL SERTT(YFL,STT,THISO,NEXTO,TRUCK(THIST,6))
       TRUCK(THIST,2) = TN + STT
       TRUCK(THIST,4)  =  NEXTO
       GO  TO  11111
C             FINAL  OUTPUT
55555 CONTINUE
C        THIS SECTION  UPDATES FULL AND MT STOCKS TO END OF SIMULATION
55556 FORMAT('  THIS IS THE FINAL OUTPUT WITH UPDATING TO END OF SIM.')
      WRITE(3,55556)
      DO 55557 E = 1,EMAX
55558 IF(OUTLET(E,7).GE.LWEEK*24*7) GO TO 55557
      OUTLET(E,5) = OUTLET(E,5) - 1
      OUTLET(E,6) = OUTLET(E,6) + 1
      OUTLET(E,22)  =  OUTLET(E,22) + 1
      IX = IY
      CALL RANDU(IX,IY,YFL)
      CALL SERTO(YFL,OUTLET(E,4),OUTLET(E,7),STO,MINSH,MAXSH)
      OUTLET(E,7) = OUTLET(E,7) + STO
      OUTLET(E,17) = OUTLET(E,17)+STO*((OUTLET(E,5))**2)
      OUTLET(E,25)=OUTLET(E,25)+STO*OUTLET(E,5)
      GO TO 55558
55557 CONTINUE
      CALL TOTAL(TN)
      WRITE(3,NAM2)
      CALL STATS(LWEEK,TN)
      CALL OUTPUT
      STOP
      END
```

3. COMPUTER PROGRAM

In this chapter is given a full description of the computer program written to represent the model of a truck distribution system described in Chapter 2. Figure 1 which is a flowchart of the program and Figure 2 a listing of the FORTRAN program should be read in conjuction with this chapter. Under the appropriate sub-headings is given a full description of each of the entities (OUTLETS, ROUTES and TRUCKS), the method of timing and the input/output routines of the program. Also given is a brief discussion on the choice of programming language.

3.1 PROGRAMMING LANGUAGE.

When writing a simulation program it is possible to use either a general purpose language (FORTRAN, ALGOL, etc.) or a simulation language (C.S.L., GPSS) (Ref. 5, 11) As this program was being written for the IBM 360 and C.S.L. is not available on this machine, the choice of special purpose programming language was restricted to GPSS. GPSS (General Purpose System Simulation) is an extremely powerful simulation language and quite easy to program. A particular advantage is the ease of collecting statistics, and taking samples of statistical distributions. There is a disadvantage in GPSS, however in that it is oriented to problems where there is a flow through fixed stations, with the objects moving through the stations being created as they arrive and destroyed when they leave. As examples consider ships moving in and out of a port, or parts flowing along an assembly line. The truck dispatching problem, however, is a closed cycle where trucks cycle from depot to outlets to depot and containers follow the same line. When considered in this manner the truck dispatching problem is not ideally suited to simulation by GPSS.

General purpose languages have the advantage of great flexibility, but the disadvantage of being slightly more difficult to use. For this problem, however, the flexibility offered by a general purpose language outweighed the disadvantages, particularly as it was intended to write a program that could easily be adapted to special requirements. (In chapter 4 several suggestions are made of how this program can be altered to suit specialised problems). Among the general purpose languages FORTRAN is the most universal and for this reason it was chosen for this program. During the course of this study the choice was shown to be well justified.

3.2 OUTLETS.

In the program each of the outlets is treated as a separate entity which is a member of the total set of outlets. Each outlet has a number of attributes and in the program these are held in an array OUTLET, one row of the array giving all the attributes of a particular OUTLET, each column representing a different attribute. For the program written, this array has a fixed size and hence the number of outlets that can be used is limited, however, to increase the capacity of the program it is only necessary to alter the dimension statements at the beginning of each subroutine. Each outlet can therefore be identified by its row number and a particular attribute by the appropriate column number, thus OUTLET (6, 5) refers to the number of full containers held in outlet number 6. A full description of each of the attributes is given below after the number of the column of the array in which it is found.

1. The code number of the outlet, this is also equal to the row number or position in the array OUTLET.

2 and 3.
The "X" co-ordinates respectively of the outlet's location. These two attributes are used in calculating the distance between two adjacent outlets on a route and hence/

hence/ the time a truck takes to travel between them. They are found by plotting all the outlets in a map and reading of the "X" and "Y" co-ordinates relative to some axes.

4. The annual consumption of containers by the outlet. This is an average value used in calculating the time for a container to be consumed.
5. The number of full containers held in the outlet.
6. The number of empty containers held in the outlet.
7. The time at which the next container will be emptied (or consumed).
8. The maximum number of full plus empty containers held by the outlet at any one time.
9. The time at which the maximum stock of full plus empty containers occurred.
10. The minimum number of full plus empty containers held by the outlet at any one time.
11. The time at which the minimum number of full plus empty containers occurred.
12. This attribute is used by the program in calculating the standard diviation of the number of containers held by the outlet during the simulation. Its value is determined by accumulating in this column the square of the number of full plus empty containers in stock at any one time, multiplied by the time for which this stock occurred. That is if at time t_1
    OUTLET (I, 12) = X
    OUTLET (I, 5) + OUTLET (I, 6) = Y, the total stock of the outlet; and the stock remains constant until time t_2 then at time t_2.
    OUTLET (I, 12) = $X+Y^2 . (t_2 - t_1)$.
13. The maximum number of full containers in stock at any one time.
14. The time at which the maximum stock of full containers occurred.

15. The minimum number of full containers in stock at any one time.

16. The time at which the minimum stock of full containers occurred.

17. A figure similar to that in column 12 used for calculating the standard deviation of the full stock during the simulation.

18. The number of deliveries made to the outlet during the simulation.

19. The number of containers delivered to the outlet during the simulation.

20. The number of containers to be delivered at the next delivery. This is the outlet's "order".

21. The "ideal" full stock at the outlet. This is a figure which has been determined as the ideal stock for the outlet to prevent it running out of containers between scheduled deliveries. It would normally contain a safety factor to cover the possibility of strikes by truck drivers etc.

22. The number of containers consumed by the outlet during the simulation.

23. The time at which the last delivery occurred.

24. A figure similar to that in column 12 for determining the mean stock of full plus empty containers in the outlet during the simulation. Using the figures given above, if at t_1 OUTLET (I, 24) = Z then at t_2 OUTLET (I, 24) = Z + Y. $(t_2 - t_1)$.

25. A figure similar to that in column 24 for determining the mean stock of full containers in the outlet during the simulation.

Of the above attributes, 1 is determined automatically by the program and is used in checking the input data. Attributes 2, 3, 4 and 21 are determined external to the program and are part of the initial input.

The remaining attributes are initally set at zero and from then on automatically adjusted by the program. Attributes 5, 6, 7, 20 and 23 are solely for use by the program in making calculations while attributes 8 - 19, 22, 24 and 25 are used when the simulation is completed to provide statistics on the operation of the model.

For convenience, the depot is also treated as an outlet, being located in the row immediately following the last outlet, however, the only significant attributes are 1, 2 and 3, the remainder not being used.

To allow the collection of statistics on the total number of containers in the field, a further row is added to the matrix. At the end of each day the total number of full containers in outlets is found (by totalling the values of attribute 5 for each of the outlets) and this is entered in column 5 of this extra row. Similarly the total number of empty containers is entered in column 6. These two figures are then used in calculating corresponding statistics for the other columns in this row. By this means statistics are available on the total number of containers in the field. The calculation of these statistics is carried out by a subroutine TOTAL.

3.3 TRUCKS.

The program treats the entities trucks in a similar way to outlets. That is, a row for each truck in the array TRUCK, each column representing a different attribute of the TRUCK. The column of the various attributes is given below.

1. The code number of the truck, which is equal to its row number in the array.
2. The time at which the truck's next delivery will be made.
3. The code number of the route being followed by the truck.
4. The code number of the next outlet to receive a delivery from the truck.

5. The number of full containers on the truck.
6. The number of empty containers on the truck.
7. The time at which the truck commenced the route being followed.
8. The total time the truck has spent following routes other than the one it is presently following.
9. The maximum time the truck has spent to complete any one route.
10. The code number of the route taking the longest time to complete.
11. The minimum time taken by the truck to complete any one route.
12. The code number of the route taking the minimum time to complete.
13. The number of routes completed by the truck during the simulation.
14. This attribute is used to determine the standard deviation of the times taken by the truck to complete routes. It is the sum of the squares of the time taken to complete each route.
15. The capacity of the truck given in terms of the maximum number of containers that can be carried.

Attribute 1 is determined by the program in making calculations and the remainder are used for generating statistics on the performance of individual trucks and the truck fleet as a whole, at the end of the simulation.

3.4 ROUTES.

While the routes do not correspond to physical objects in a real distribution system as do the trucks and outlets, they are treated in a similar way by the program. That is an array ROUTE is formed, each row corresponding to a different route and each column being an attribute of the route.

This array ROUTE is really the centrepiece of the program as it gives the sequence in which deliveries are made and hence determines the timing of the trucks and when the stock of the outlets will be updated. The following are the column numbers of the array ROUTE and the attributes represented by each of these columns.

1. The code number of the route which is equal to the row number in the array.
2. The code number of the truck following the route at a particular time.
3. The time at which deliveries on this route were commenced by the truck in column 2.
4. The number of times the route has been used or followed during the simulation.
5. The total time spent following this route during the simulation.
6. The number of outlets on the route.
7. The column number in which is to be found the code number of the next outlet on the route to receive a delivery.
8. The number of containers to be delivered along the entire route.
9. The day of the week this route is used. This is a number in the range 0 - 6 representing the days of the week.

Columns 10 onward contain the code numbers of the outlets to receive deliveries on this route and are in the order in which they receive deliveries. This information is read in as part of the input data.

Attribute 1 is automatically determined by the program and used to check the input data. Attributes 6, 9 and 10 onwards are part of the input data, while attributes 2, 3, 7 and 8 are used by the program in carrying out calculations. Attributes 4 and 5 are used for determining statistics on the/

the/ time taken to complete the route.

The program automatically inserts the depot as the last outlet on the route, and at the same time increases by one (1) the value in column 6 to allow for this.

3.5 TIMING AND SEQUENCING.

Perhaps the single most important thing influencing the success of a simulation study is the method of timing and sequencing used. For this reason the methods used in this study will be described in some detail in this section (Ref. 7, 12).

3.5-1 TIME FLOW MECHANISM.

There are two methods of handling the flow of time in a digital computer simulation - fixed time increment or variable time increment (Ref: 7, 12). With the fixed time increment method a "clock" is simulated which indicates the instant of real time that has been reached in the simulation, in this program this is called TN (Time Now). The clock is updated by fixed increments and after each increment the system is scanned to determine any events that have occured in the interval since the last increment. These events are then recorded by the program and another time increment made. With the variable time increment method the system is scanned to determine the next event to occur and the clock is advanced to the time of this event. The event is recorded and the system examined again for the next event. This process is repeated to the end of the simulation.

With both methods there are advantages for particular applications. Fixed time increment methods are particularly useful when events occur at fixed time increments (corresponding of course to the time increment used in advancing/

advancing/ TN), or in studies of continuous systems where there are no major events to be studied such as in the simulation of a servo-mechanism. There is an advantage in using the fixed time increment method in that programming is relatively simple as the program consists of a simple loop of increasing time and adjusting the status of each entity. Another advantage is that it is relatively easy to estimate the computer time required to simulate a particular period of real time. Variable time increments are of advantage when:-

1. Events do not occur at fixed time intervals and it is unnecessary to update the status of the entities between events - under these circumstances computer time is saved as updating only occurs when an event occurs.
2. The sequence of events is important. In this situation an event can effect other succeeding events so it is important that the events are handled in strict sequence. In the fixed time increment method it is possible for two events to occur in the same time interval with no means of determining which should be treated first.

For this program variable time increments were chosen mainly to reduce computer time, as there are reasonably large intervals of time between events and so many fruitless searches for current events are avoided. It would, however, have been quite satisfactory to use a fixed time increment method.

3.5-2 CALCULATION OF TIME MOVEMENT.

Only two types of event cause the "clock" to be advanced, a truck reaching an outlet to make a delivery and the beginning of a day.

The method followed is that column 2 of the array TRUCK is scanned to determine which truck will next make a call (this truck is then labelled by the program THIST).

TN (Time Now) is then made equal to TRUCK (THIST, 2). TRUCK (THIST, 4) gives the outlet receiving this delivery. If the time at which this outlet is next due to consume a container is less than TN then the status of this outlet is updated and the time for the next container to be consumed there is calculated. This new time is compared with TN and the above process is repeated. When the time at which the outlet willnext consume a container becomes greater than TN then the number of full containers on the truck is reduced by the order quantity for the outlet and the number of full containers in the outlet is increased by a similar quantity. In a similar method the empty containers in the outlet are transferred to the truck; if the truck has insufficient capacity, some of the empty containers are transferred. Throughout this process the program is automatically changing any of the other attributes of the trucks or outlets which are altered by the consumption or transferring of containers.

When the delivery has been completed a search is made to find the next outlet to receive a delivery by the truck. The time for the truck to reach this outlet and make a delivery is determined and recorded in TRUCK (THIST, 2). A search is then made to determine the truck next to make a delivery and the above process is repeated. As each truck returns to the depot a check is made to see if anymore routes remain to be serviced on that day. If they do the truck is assigned to one of the remaining routes, if not the time for the next call for the truck is advanced to midnight. After all the trucks have returned to the depot every value in column 2 of the array TRUCK will equal midnight. When the check is made to determine the next truck to make a delivery, the time of this delivery is examined to determine if it is midnight. If it is, the beginning of day routine is entered.

The first thing done by the beginning of day routine is to check whether the new day is Saturday or Sunday and if so it advances the time to Monday morning. (The program assumes no deliveries are made on Saturdays and Sundays, but it would be a simple matter to alter the program to change this). Having done this the stock of each of the outlets receiving a delivery on the new day is updated to 5.00 p.m. the previous day and this stock compared with the outlets' ideal stock and an order made of the difference. The basis of this is that the person running the outlet would determine his order by what he estimates his stock will be at the end of the day prior to the delivery.

Trucks are then assigned to each route and the time taken till the trucks complete their first delivery is calculated and entered as an attribute of the truck. If there are insufficient trucks available to handle all routes the number of the truck assigned to the remaining routes is set to zero and this is used when the trucks return to the depot to determine which routes still require servicing. A search is then made to determine the truck next requiring delivery and the above process is repeated until the simulation is completed.

The simulation is completed when TN reaches a value determined by the person running the simulation and included in the input data. A check is made to see if the simulation has been completed at the beginning of each day.

3.5-3 CALCULATION OF DELIVERY AND CONSUMPTION TIMES.

As stated above the times to make a delivery or consume a container are considered to belong to some standard statistical deistribution. The type of distribution and the parameter of the distribution will, however, depend largely on the real system being studied.

For this reason the calculation of these times is carried out by four subroutines, RANDU, SERTT, SERTO and INPUT, so that to alter the program to suit any special circumstances it is only necessary to alter one or more of these subroutines.

The subroutine INPUT is called once in the program, immediately after the main input has been read. Its purpose is to read in the parameters of the statistical distributions being used in the calculation of the times to make deliveries and consume containers.

RANDU is a standard IBM 360 subroutine for the calculation of pseudo random numbers and is used to sample the statistical distribution of delivery and consumption times.

The time for a container to be consumed is calculated by the subroutine SERTO (SERvice Time of Outlet). It is assumed in this program that time taken for a container to be consumed is uniformly distributed about the mean time. The mean time is determined by calculating the average time to consume a container from the outlet's average annual sales. The first card read by subroutine INPUT is a card of type number 8, (see section 3.6), in the first field of which is the percentage variation about the mean consumption time. That is if card type 8 contains 20.00 in the first field it means that the time taken to consume a container varies uniformly between 80% and 120% of the mean time. To calculate the time taken for a container to be consumed a pseudo random number is generated by RANDU and this is then used to sample the distribution generated from the average annual consumption of the outlet and the range of the distribution read in by subroutine INPUT.

The calculation of delivery times is very similar to that of consumption times and is carried out by subroutine SERTT (SERvice Time of Truck).

Card of type number 9 read by subroutine INPUT contains in the first field the range of the uniform distribution of the delivery times. Card of type number 10 contains in the first field the average number of minutes taken by a truck to cover one mile and in the second field the average number of minutes taken by a truck to cover one mile and in the second field the average number of minutes to deliver a container and in the third field the average number of minutes taken during a delivery to check dockets etc. Therefore when SERTT is called it first calculates the distances to be travelled by the truck in making the next delivery and the average time taken to do this. From the number of containers delivered it then calculates the average time to make the delivery, including the process of filling in the appropriate dockets etc. A pseudo random number generated by subroutine RANDU is then used to sample the distribution having as a mean the average total time to make this delivery (driving time, and the time to transfer containers and fill out any paper work) and the range read in by card of type number 9. The resulting figure is then taken as the time required to make the delivery.

The uniform distribution was used in this case because of its simplicity. In a study of a real distribution system it would be necessary to carry out a work study of the truck dispatching and determine the actual statistical distribution appropriate to the system. With this data it would then be a relatively simple matter to rewrite the subroutines INPUT, SERTT and SERTO to suit this situation.

3.6 INPUT DATA

The format of the input data was designed to allow its preparation to be fairly simple while containing a number of error checks.

Each card consists of sixteen (16) fields of five (5) characters, not all of which need be significant to the program. The last field used on the card is a type number and shows the type of data contained on the card, there are ten (10) types of card including types 8, 9 and 10 described in the last section. Each card type 1 - 7 is described below. See the Appendix for detailed format of each of the cards.

Type 1. There is only one card of type 1 required for any simulation and it is the firstcard of the input deck. The information contained on this card is the number of routes, trucks and outlets to be considered in the simulation. When this card is read checks are made that the type number is correct and that the arrays ROUTE, TRUCK and OUTLET are large enough to handle the problem.

Type 2. This is the second card read and again there is only one of this type of card required for a simulation. The information contained on this card is:-

(a) the duration of the simulation.
(b) the earliest hour of the day at which outlets can consume containers (equivalent to shop opening time).
(c) the latest hour of the day containers can be consumed (closing time).
(d) the number of days per week over which containers can be consumed.
(e) the earliest time deliveries can be made.
(f) the latest time deliveries can be made.
(g) the number of days per week that deliveries can be made.

Type 3. There is a card of type 3 for each route in the simulation and all cards of type 3 immediately follow the card of type 2 and are placed in order of ascending route number.

As well as the route number and card type number which are used as input checks this card also contains the number of outlets on the route and the day of the week on which/

which/ this route is used.

Type 4. Again there is a card of type 4 for each route, sorted in route number order and these immediately follow the cards of type 3. These cards as well as input checks contain the code numbers of the outlets on this route in the order in which deliveries are made.

Type 5. There is one card of type 5 for each truck and these are sorted in ascending order of truck number and immediately follow the cards of type 4. Apart from input checks these cards contain the capacity of the trucks.

Type 6. For every outlet there is a card of type 6, these are also sorted in order of outlet number and follow the cards of type 5. The data contained on these cards is:-

c

(a) "X" co-ordinate of location.
(b) "Y" co-ordinate of location.
(c) average annual consumption.
(d) initial stock of full containers.
(e) initial stock of empty containers.
(f) the ideal full stock for the outlet.

Type 7. This card gives the "X" and "Y" co-ordinates of the depot and immediately follows the cards of type 6.

Cards of type 8, 9 and 10 follow the cards of type 7 and are as described in the previous sub-section.

As each card is read a test is made to check whether it is in its correct sequence and whether the data contained is not conflicting or outside the scope of the program. If errors are detected these are recorded on the printout and when the whole of the input data has been read the program is halted to allow the cards to be corrected.

3.7 OUTPUT DATA.

There are three types of output produced by the program.

THE FOLLOWING PAGES GIVE FULL DETAILS OF THE SIMULATION AND RELEVANT RESULTS AND STATISTICS

```
DETAILS
          NUMBER OF WEEKS SIMULATED      =   1
          NUMBER OF OUTLETS CONSIDERED   =   6
          NUMBER OF ROUTES CONSIDERED    =   2
          NUMBER OF TRUCKS CONSIDERED    =   2

RESULTS
          TOTAL NUMBER OF UNITS CONSUMED  =        42.
          TOTAL NUMBER OF UNITS DELIVERED =        36.
          AVERAGE FULL STOCK IN OUTLETS   =        43.
          AVERAGE TOTAL STOCK IN OUTLETS  =        56.

          AVERAGE DELIVERY TIME PER ROUTE =   4.500
          AVERAGE DELIVERY TIME PER UNIT  =   0.250

          MAXIMUM TOTAL STOCK IN OUTLETS  =        60.  TIME OF MAX = WEEK #   0 DAY # 0
```

R O U T E   S T A T I S T I C S

ROUTE NUMBER	AVERAGE TIME PER ROUTE
1	4.000
2	5.000

SIMULATION OF ROUTE STATISTICS

FIG. 3

TRUCK STATISTICS

TRUCK #	AV. TIME/ROUTE	AV. HOURS/DAY	MAX TIME/ROUTE	MAX ROUTE #	MIN TIME/ROUTE	MIN ROUTE #	ROUTE TIME S-D
1	4.500	1.125	5.000	2.	4.000	1.	0.500

```
IHC210I PROGRAM INTERRUPT(P) OLD PSW IS   FF25000FA2010DCC
IHC210I PROGRAM INTERRUPT(P) OLD PSW IS   FF25000FA2010DE8
IHC210I PROGRAM INTERRUPT(P) OLD PSW IS   FF25000FA2010DF0
```

Note. These are caused by a "divide by zero" due to the nature of the artificial data used.

TRUCK #	AV. TIME/ROUTE	AV. HOURS/DAY	MAX TIME/ROUTE	MAX ROUTE #	MIN TIME/ROUTE	MIN ROUTE #	ROUTE TIME S-D
2	0.0	0.0	0.0	0.0	*******	0.0	0.0
ALL TRUCKS	4.5	0.563					0.500

TRUCK STATISTICS

FIG. 4

OUTLET STATISTICS

OUTLET NUMBER	MAX TOTAL	TIME MAX TOT	MIN TOT	TIME MIN TOT	MAX FULL	TIME MAX FULL	MIN FULL	TIME MIN FULL	AV TOT	AV FULL	S-D TOTAL	S-D FULL	# OF DELIVS	UNITS DELIV	UNITS CONSUM
1	10.	0.0	9.	107.	10.	0.0	5.	96.	6.	2.	5.0	6.6	1.	8.	7.
2	10.	0.0	9.	108.	10.	0.0	5.	96.	6.	2.	5.0	6.6	1.	8.	7.
3	10.	0.0	9.	59.	10.	0.0	7.	48.	3.	4.	4.6	5.6	1.	4.	7.
4	10.	0.0	9.	60.	10.	0.0	7.	48.	3.	4.	4.6	5.6	1.	4.	7.
5	10.	0.0	9.	61.	10.	0.0	7.	48.	3.	4.	4.7	5.6	1.	4.	7.
6	10.	0.0	9.	62.	10.	0.0	7.	48.	5.	4.	4.8	6.5	2.	8.	7.
ALL	60.	48.	54.	192.	60.	0.0	36.	192.	56.	43.	2.4	7.0	7.	36.	42.

FINAL STATUS OF ROUTES

OUTLET STATISTICS

FIG. 5

The first is an error report on the input data as explained in the previous section, the second is the current status of the simulation and the final type of output contains statistics of the simulation.

A subroutine OUTPUT has been written that will print out the arrays ROUTE, TRUCK and OUTLET in an easily read form. A call statement for this subroutine can be inserted anywhere in the program to give the current status of the simulation. This is normally used only at the beginning and end of the simulation to allow a check to be made on the input data and the final status of the outlets. However it can be of use in giving diagnostics if problems are being incurred in running the program, for example, if the trucks are being continually overloaded a CALL OUTPUT statement can be inserted at the end of the beginning of day routine to determine when the overloading is occuring.

The subroutine STATS is used at the end of the program to give a summary of the simulation and to provide sufficient statistics on the operation of the model to allow it to be evaluated. Figures 3, 4 and 5 show the format of this output. The statistics provided are mainly means and standard deviation of times taken for various operations and numbers of containers being used. These figures give an indication of the efficiency of the model being studied and can be used to compare the likely operating costs of two models.

3.8 GENERAL.

The computer program described in this chapter was written in such a manner that it could be very easily altered by an analyst with a knowledge of FORTRAN programming. This has been accomplished by having a number of operations performed by subroutines and a minimum of cross lines within the main program.

Figure 1 shows how the mainline program has been divided into distinct blocks, each with as few entry and exit points as possible. It is therefore quite an easy matter to remove a block and replace it with another which better suits a particular application. Similarly the method of storing all the information about a particular set of entities in an array makes it a simple matter to add to the attributes of the entities if this is required. Another feature is the simple method of inputting data on the entities and the ease with which this data can be changed to obtain quite a different model.

A criticism of the program is that it is fairly inefficient and it may be worthwhile rewriting sections of it if it is desired to carry out a lengthy simulation. In particular the program might be speeded up by chaining the trucks in the sequence in which they are next to make a delivery. That is, associate with each truck an attribute which is the number of the truck which will make the call immediately following the first truck's delivery. Therefore with this feature, each time a delivery is completed this attribute will immediately show which truck is next to make a call, removing the need to search through every truck. It would of course be necessary to scan the "chain" when a delivery is completed to re-insert the truck in the "chain", but this would be more efficient than a scan of all of the trucks in the model. This was not done in this program as it would have required more complicated programming, with no benifit apart from speed, which was a farily minor consideration during the development phase.

4. APPLICATIONS AND POSSIBLE EXTENSIONS.

As suggested by the title and explained in the Introduction, the purpose of this project has been to devise a computer program that can be used to simulate a trucks dispatching system; with particular attention to the case where the product being carried by the trucks is in returnable containers owned by the distributor. The original aim had been to use the program to determine the most efficient routes to be followed by trucks, but as the project got underway many other possible applications became apparent and in some instances the program was altered to simplify its modification if necessary for further applications. This chapter summarises the possible applications and the way they can be accomplished by extending the program.

4.1 DETERMINATION OF ROUTES.

The problem for which this study was originally intended to provide a method of solution was that of determining the most efficient routes to be followed by trucks making deliveries to a number of customers. With the computer program prepared, this is accomplished by setting up hypothetical routes and then carrying out a simulation. This is then repeated for several alternative sets of routes and the results of the different simulations compared. As explained in section 3, to alter the routes it is only necessary to alter the input data cards of type number 4. The determination of the routes to be simulated can be carried out by any one of the analytical techniques referred to in the Introduction (section 1). When trying to determine the best routes, it is not essential to simulate the whole of the real system at the one time. It may be satisfactory to isolate a particular area (for example the Eastern Suburbs of Sydney) and try various combinations of routes for that area.

When the best set of routes for that particular area have been determined the process can be repeated for other areas; or it may even be possible at that stage to identify the correct method of route determination and implement new routes determined by this method without reference to a simulation study.

4.2 DETERMINATION OF OPTIMUM TRUCK SIZES.

In a similar way to the determination of optimum routes it is possible to determine the optimum truck sizes to be used. This study would most likely be carried out in conjunction with determination of routes as the number of deliveries along a route obviously influences the truck size required.

4.3 SELECTION OF FACTORY SITES.

A problem closely related to that of the dispatching of trucks is the selection of factory sites. Among the factors influencing site selection is the distribution cost of product from factory to consumer. To use this program to compare distribution costs from various sites it is simply a matter of altering to co-ordinates of the depot location to the alternate sites and running the simulation for each of these. It may, however, be necessary to choose routes relatively independent of the factory location.

4.4 TYPES OF PRODUCT DISTRIBUTED.

The program written for this study is based on the assumption that the product is delivered in returnable containers. If the product is delivered in non-returnable containers no alteration to the program is required, as it is simply a matter of ignoring empty stock figures in both the input and the output.

In cases where the product is delivered in bulk, such as petrol or gravel it is only necessary to consider the term container to refer to a unit of the product and treat the problem as for non-returnable.

The time taken for the consumption of different products will vary quite widely, so for this reason the calculation of the time to consume a container was written in a subroutine to allow simple conversion of the program to meet special circumstances. This has been dealt with at some length in section 3.5-3.

4.5 AFFECT OF A CHANGE IN SALES ON DISTRIBUTION.

A question often asked by management is:"What would happen if our sales were to increase by 'x'%." As far as the effect on the dispatching of trucks is concerned this can be quite easily answered by using this simulation program. It would only be a matter of increasing the annual consumption of each outlet or adding extra outlets(or both)and setting up new routes as necessary.

5. CONCLUSION.

The result of this project is a computer program that can quite easily be used in the study of a wide range of problems associated with the distribution of products by truck. As with all simulation programs it must be used with great care and one of the largest tasks is verifying that the model adequately represents the real system (Ref. 8).

Before using this problem a lot of work must be done to determine:-

1. The location of customers.
2. The average consumption of each customer.
3. The statistical distribution describing consumption.
4. The speed at which trucks travel and the statistical distribution describing this.
5. What are considered ideal stocks by customers.

As well as determining the above, studies must be made to determine the routes to be used and very importantly the initial conditions, or opening stocks to be assumed in each of the outlets.

When a simulation has been completed it is first necessary to examine the results to see if they agree with results obtained in the real world. Once the model has been verified it is then necessary to examine the statistics to determine the cost of a real system equivalent to the model.

Therefore, the study of a real world distribution system is a very large job, even with the use of this program. It is hoped, however, that this program can be used in the evaluation of alternate methods of truck dispatching and assist in obtaining more efficient operations in the real world.

APPENDIX

INPUT FORMAT

In this Appendix upper case letters represent variable data to be entered by the user and the lower case "b" represents a blank character. All data is entered in fields of five columns, right justified and starting at column 1, preceding zeros are not required. The format for each of the ten types of input card is as follows:-

Type 1. The format for this card is:

AAAAABBBBBCCCCCbbbb1

where AAAAA = the number of routes to be simulated.
BBBBB = the number of trucks to be simulated.
CCCCC = the number of outlets to be simulated.

Type 2. The format for this card is:

AAAAABB.BBCC.CCbbbbDEE.EEGG.GGbbbbHbbbb2

where AAAAA = the number of weeks to be simulated.
BB.BB = the hour at which consumption commences each day (shops open).
CC.CC = the hour at which consumption ceases each day (shops close).
D = the number of days per week during which consumption is carried out.
EE.EE = the earliest hour at which deliveries can be commenced during the day.
GG.GG = the latest hour at which deliveries can be made.
H = the number of days per week during which deliveries can be made.

Type 3. The format of cards of type 3 is:-

AAAAABBBBBbbbbCbbbb3

where AAAAA = the code number of the route.

BBBBB = the number of outlets on the route.

C = the day of the week that deliveries are made along this route (0 - 6, 0 = Sunday).

Type 4. The format of cards of type 4 is:-

AAAAABBBBB----------------------bbbb4

Col. 1 ... Col. 80

where AAAAA = the code number of the route.

BBBBB = the code number of the first outlet on the route.

The code numbers of the remaining outlets on the route are continued in successive fields across the card.

NOTE: No more than 14 outlets are allowed on any one route.

Type 5. The following is the format for type 5 cards:-

AAAAABBBBBbbbb5

where AAAAA = the code number of the truck.

BBBBB = the number of containers that can be carried by the truck.

Type 6. The following is the format of type 6 cards:-

AAAAABB.BBCC.CCDDDDDEEEEEGGGGGHHHHHbbbb6

where AAAAA = the code number of the outlet.

BB.BB = the "X" co-ordinate of the outlet's location.

CC.CC = the "Y" co-ordinate of the outlet's location.

DDDDD = the average annual consumption of containers by the outlet.

EEEEE = the initial stock of full containers.

GGGGG = the initial stock of empty containers.

HHHHH = the ideal stock of full containers.

Type 7. Cards of type 7 have the following format:-

AAAAABBBBBbbbb7

Where AAAAA = the "X" co-ordinate of the depot's location.

BBBBB = the "Y" co-ordinate of the depot's location.

Type 8. The format for cards of type 8 is:-

AA.AAbbbb8

where AA.AA = the percentage spread on either side of the mean time to consume a container.

Type 9. For cards of type 9 the format is:-

BB.BBbbbb9

where BB.BB = the percentage spread about the mean time to make a delivery.

Type 10. The cards of type 10 have the following format:-

AA.AABB.BBCC.CCbbb10

where AA.AA = the number of minutes taken for a truck, on average, to drive one mile.

BB.BB = the average number of minutes taken to deliver one container from truck into outlet.

CC.CC = the average number of minutes taken by "overhead" in making a delivery.

REFERENCES.

1. CLARKE, G. & WRIGHT, J.W. "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points." J.O.R.S.A. 12, 4 (1964) pp. 568 - 581.
2. DANTZIG, G.B. & RAMSER, J.H. "The Truck Dispatching Problem". Management Science, (1959) pp. 80 - 91.
3. EVANS, G.W. & WALLACE, G.F. & SUTHERLAND, G.L. "Simulation Using Digital Computers". Prentice-Hall Inc. (1967).
4. GASKELL, T.J. "Bases for Vehicle Fleet Scheduling". Operational Research Quarterly 18, 3 (1967) p. 281.
5. KIVIAT, P.J. "Development of New Digital Simulation Languages". The Journal of Industrial Engineering 17, 11 (1966) p. 604.
6. KNIGHT, K.W. & HOFFER, J.P. "Vehicle Scheduling with Timed and Connected Calls: A Case Study". Operational Research Quarterly 19, 3 (1968) p. 299.
7. LAVE, R.E. "Timekeeping for Simulation". The Journal of Industrial Engineering 18, 7 (1967) p. 389.
8. NAYLOR, T.H. & FINGER, J.M. "Verification of Computer Simulation Models". Management Science 14, 2 (1967).
9. NAYLOR, T.H., BALINTFY, J.L., BURDICH, D.S. & KONG, CHU "Computer Simulation Techniques". John Wiley and Sons, Inc. (1968).
10. SUSSAMS, J.E. "Some problems Associated with the Distribution of Consumer Products". Operational Research Quarterly 19, 2 (1968) p. 161.
11. TOCHER, K.D. "Review of Simulation Languages". Operational Research Quarterly 16, 2 (1965) p. 189.
12. TOCHER, K.D. "The Art of Simulation". D. van Nostrand Co. (1963).
13. WENTWORTH, F.R.L. "Route Planning by Computer". Data Processing March - April 1967, p. 106.