NBSIR 85-3178

Paratransit Advanced Routing and Scheduling System Documentation: Routing and Scheduling Dial-A-Ride Subsystem

Howard K. Hung
William G. Hall
Robert E. Chapman

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Applied Mathematics
Gaithersburg, MD 20899

July 1985

Sponsored by
U.S. Department of Transportation
Urban Mass Transportation Administration
Office of Technical Assistance
Washington, DC 20590

NBSIR 85-3178

PARATRANSIT ADVANCED ROUTING AND SCHEDULING SYSTEM DOCUMENTATION: ROUTING AND SCHEDULING DIAL-A-RIDE SUBSYSTEM

Howard K. Hung
William G. Hall
Robert E. Chapman

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Applied Mathematics
Gaithersburg, MD 20899

July 1985

Sponsored by
U.S. Department of Transportation
Urban Mass Transportation Administration
Office of Technical Assistance
Washington, DC 20590

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

PREFACE

This research was conducted under the sponsorship of the Paratransit Integration Program (PTIP), Urban Mass Transportation Administration, U.S. Department of Transportation, by the Operations Research Division, Center for Applied Mathematics, National Engineering Laboratory, National Bureau of Standards. Mr. Edward G. Neigut was the UMTA program manager for the project. The Routing/Scheduling Dial-A-Ride (RSDAR) algorithm described in this report creates tour schedules from a file of trip requests made by dial-a-ride patrons. A trip request consists of the number of passengers (a patron and his party), a pickup location, a dropoff location and either a desired pickup or dropoff time. Tour schedules created by RSDAR contain a list of locations, times of vehicle stops, and the names of patrons boarding and alighting at each stop.

The RSDAR is a heuristic algorithm designed for operation in a batch mode. It can accommodate up to eight different characteristics for a patron, and up to 15 classes of priority. Conditions such as separate limits for the number of wheelchairs on a vehicle and economies of scale on patron dwell time are also included. The RSDAR is intended for use by dial-a-ride system operators in routing and scheduling their daily trip requests.

This report is designed as a technical reference for setting up and maintaining the RSDAR model. However, a preprocessor, CONENV, which constructs physical and policy enivronments, and contains trip request information; and a post processor, GREPOR, which generates hard copy of all necessary reports, are integral parts of the Advanced Routing and Scheduling System. Hence, a reader should read the reports describing CONENV and GREPOR, and the report on data structure before beginning to implement the system.

ABSTRACT

The Advanced Routing and Scheduling System (ARSS) is a software system designed to route and schedule patrons in a dial-a-ride environment. The system consists of three subsystems: CONENV, a preprocessor, constructs physical and policy environments; RSDAR routes and schedules patrons; and GREPOR generates hard copy of all necessary reports. This report only describes RSDAR.

The RSDAR is a heuristic algorithm. It assigns patrons to form subtours in time intervals; these subtours are then linked to become a tour.

Patrons are included in a subtour on the basis of best use of the remaining time of the base trip. Subtours are included in a tour on the basis of a measure of best productivity.

The model is written in FORTRAN and complies with the American National Standards Institute X3.9-1978 standard for that language.

Keywords: Dial-a-ride; heuristic algorithm; routing; scheduling

Table of Contents

Page

Preface.. iii

Abstract... iv

List of Figures.. vii

List of Tables... viii

1. Introduction.. 1

2. A Heuristic Routing/Scheduling Dial-A-Ride (RSDAR) Algorithm... 6
   2.1 Basic Concepts.. 6
      2.1.1 Tour, Subtour, Trip and Base Trip.................... 6
      2.1.2 Service Quality Constraints.......................... 7
      2.1.3 Pickup and Delivery Time Windows..................... 7
      2.1.4 Priority... 10
      2.1.5 Productivity Measure................................. 10
      2.1.6 Dwell Time... 11
      2.1.7 Vehicle Capacity..................................... 12
      2.1.8 Mobility Code.. 12
   2.2 Heuristic Approaches...................................... 13
      2.2.1 Time Interval.. 13
      2.2.2 Maximum Remaining Time............................... 14
      2.2.3 Productivity Measure................................. 14
      2.2.4 Base Trip Matrix..................................... 15
      2.2.5 Compatibility Matrix................................. 16
      2.2.6 Potential Upper Bound................................ 16
      2.2.7 Time and Locations of Tour Start, End, and Break... 17
      2.2.8 Shifting... 18
      2.2.9 Call Back.. 19
   2.3 Flow Chart.. 19

3. Functional Description of RSDAR.............................. 21
   3.1 The Computer Program...................................... 21
      3.1.1 Subroutine MAIN...................................... 21
      3.1.2 Subroutine MNTOUR.................................... 24
      3.1.3 Subroutine SBTOUR.................................... 25
      3.1.4 Subroutine SBTGEN.................................... 25
      3.1.5 Subroutine TRCOMB.................................... 25
      3.1.6 Subroutine FESIBL.................................... 25
      3.1.7 Subroutine RMTIME.................................... 25
   3.2 Input Data Requirements................................... 26
   3.3 Parameters.. 28
   3.4 Model Output.. 30

4. Operational Characteristics of RSDAR........................... 32
4.1 Computational Results with RSDAR........................... 32
4.2 Computer System Configuration........................... 37
4.3 Computer Hardware and Software Requirements........................... 38

Appendix A: Internal Data Formats........................... A1

Appendix B: Input Formats........................... B1

Appendix C: Interface Formats........................... C1

Appendix D: Output Formats........................... D1

Appendix E: Sample Input........................... E1

Appendix F: Sample Interfaces........................... F1

Appendix G: Sample Output........................... G1

Appendix H: Subroutine Specifications........................... H1

Appendix I: Program Listings........................... I1

Appendix J: Test Data........................... J1

LIST OF FIGURES

		Page
1.1	Schematic View of ARSS	5
2.1	Time Windows for Pickup Specified and Dropoff Specified Trips	9
2.2	The Relative Position of Dwell Time to Scheduled Pickup Time and Dropoff Time	12
2.3	Flow Diagram of the RSDAR Algorithm	20
3.1	Flow Diagram of Subroutines of RSDAR	23

LIST OF TABLES

Page

4.1 Results of #1 Test Series (Using Productivity Measure #1 for All Patrons. 34

4.2 Results of #2 Test Series (Using Productivity Measure #1 for Wheelchair Patrons and Productivity Measure #3 for Regular Patrons. 34

4.3 Tour Assignments of #1 Test Series 35

4.4 Tour Assignments of #2 Test Series 36

4.5 Test Results of Different Number of Priorities 37

1. INTRODUCTION

Many different types of paratransit systems have been in operation for over a decade. However, most of them have been manually routed and scheduled. The first major effort to develop a computer aided routing and scheduling approach for demand responsive transit was the CARS (Computer Aided Routing System) research project at the Massachusetts Institute of Technology from 1967 to 1971. Subsequently, a field demonstration was implemented in Haddonfield, New Jersey from 1972 to 1975. During most of the demonstration period, routing and scheduling of patrons were done manually; computer dispatching was used towards the end of the project. The computer control algorithm used in Haddonfield was that developed at MIT during the CARS project. A second field demonstration was implemented in Rochester, New York from 1975 to 1978. The computer control system used in Rochester was related to the system developed in the CARS project, although modifications were made to reflect the experience gained in Haddonfield. Immediate assignment of all patrons requesting service is a common feature of these two computer control systems. The assignment not only selects a vehicle, but also inserts the new pickup and dropoff stops in the tour of a vehicle.

A routing and scheduling algorithm for patrons with advance reservations was developed for the Urban Mass Transportation Administration at the University of Maryland. The algorithm was a mathematical programming approach, but embedded with heuristics.

When UMTA asked NBS to undertake the current project of developing a dispatching, scheduling, and routing algorithm and supporting operational software packages, it specified the following six basic guidelines to be observed:

(1) Products must be user oriented.

(2) Solution techniques must be simple.

(3) Products and documentation must be structured to facilitate the separation and use of individual, self-contained segments.

(4) Programs must be portable.

(5) Programs must be executable on low cost minicomputers.

(6) Software packages must be amenable to assembly and/or integration into systems at sites to suit the specific needs of those localities.

In consideration of the above mentioned six guidelines, NBS designed a software package with three stand-alone, but related, subsystems. The first of these subsystems, CONENV, constructs the physical environment and the policy environment; reads, transforms and augments trip request information; and creates required environmental reports. The physical environment includes vehicle characteristics such as initial location, final location, vehicle capabilities, capacities and dwell times; it defines geographic zones and zone-to-zone speeds. The policy environment includes service time intervals, policy-dictated breaks in service, definition of priority for both patron and vehicle accommodation capability, and definition of service quality requirements.

The trip request input is range-checked, transformed, and augmented by required environment information. The checking is extensive: in most cases, a non-legitimate entry is corrected to a default value and processing continues. When a default is impossible, the trip is omitted. Any abnormality generates a warning message.

Mandatory outputs are information (stored internally) necessary for the execution of subsequent subsystems and monitoring/error information (CRT and hard copy). There are also a number of independent and optional hard copy reports.

After reading file zc.rsd, output of the first subsystem, (i.e., CONENV), the second subsystem, RSDAR, performs the routing and scheduling of all trip requests. After routing and scheduling are completed, RSDAR writes results on file zr.gre. They are used as input to the third subsystem, GREPOR. The algorithm for the RSDAR subsystem is a collection of heuristics which will be described in detail later in this report.

The third subsystem, GREPOR, creates the necessary hard copy reports using as input the output from CONENV and RSDAR as well as the input file zs.gre. There are two types of report. The first is the driver trip list, designed specifically for the vehicle operator. The second, the dispatcher report, has considerably more information than the driver requires and is intended for the dispatcher/scheduler. Both report types are generalized to allow for user specified item selection, sorting, and paging.

Figure 1.1 depicts the relationship between the three subsystems with respect to input, output, and interfaces. This report is one of a sequence of reports, focusing on the RSDAR subsystem of the Advanced Routing and Scheduling System (ARSS).

The first report of the sequence, NBSIR 85-3174, Paratransit Advanced Routing and Scheduling System Documentation: Functional Program and Data Specifications, establishes the concepts upon which the software development was based. The overall tone of that report is general; it does not address program and data organization details such as formats and specific computations. It describes the total paratransit environment and defines the functions, ranges and accuracy requirements.

The remaining reports of the series describe the software developed. In accordance with the aforementioned guidelines, each report is independent and self contained. There is redundancy among the reports since each is intended to be definitive with respect to a specific component or aspect of the system and only descriptive of the remainder.

The series of reports:

1 - NBSIR 85-3175 PARASSD:* USER/OPERATOR MANUAL

2 - NBSIR 85-3176 PARASSD: DATA SPECIFICATION

3 - NBSIR 85-3177 PARASSD: ENVIRONMENT CONSTRUCTION SUBSYSTEM

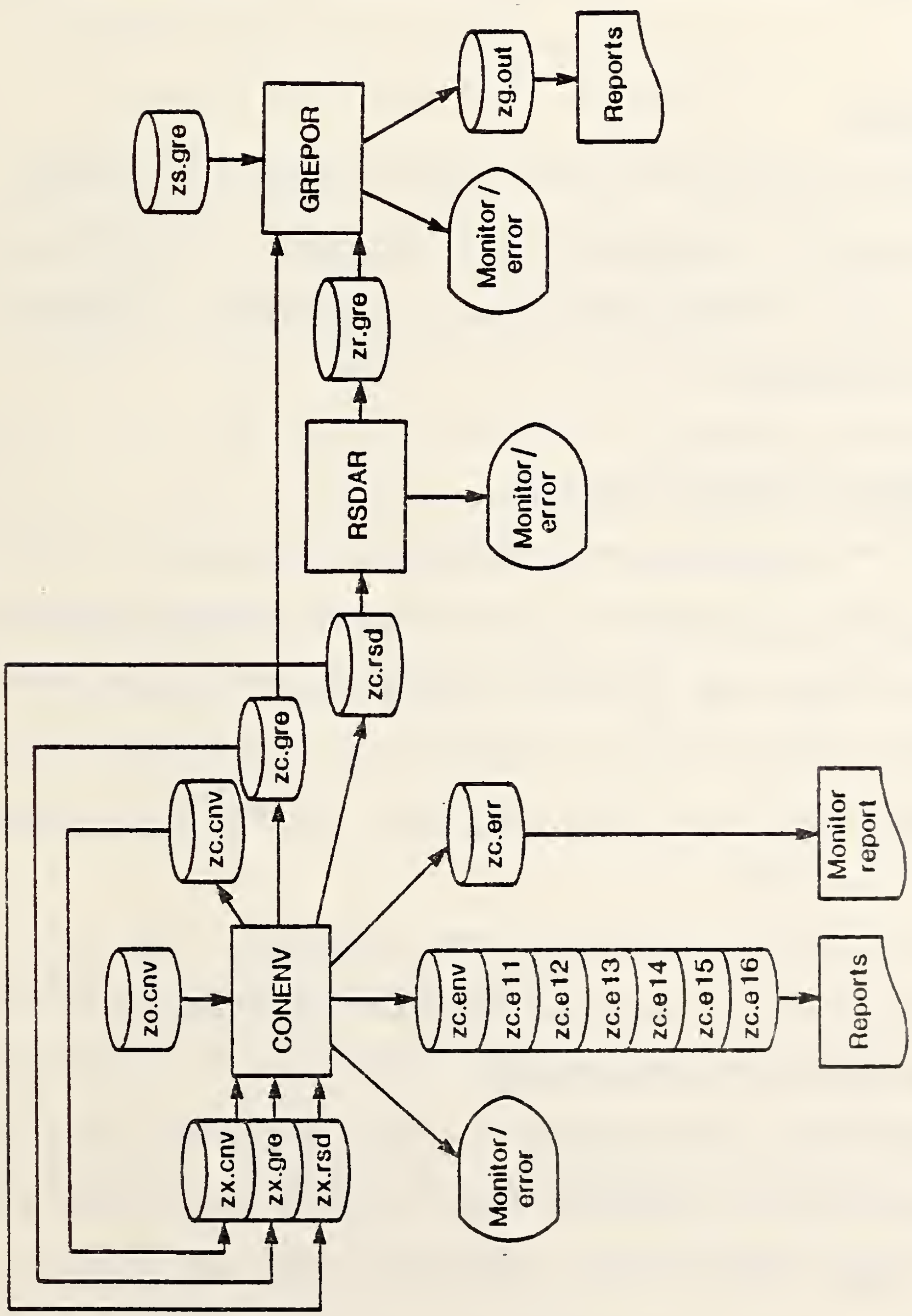
4 - NBSIR 85-3178 PARASSD: ROUTING AND SCHEDULING DIAL-A-RIDE SUBSYSTEM

5 - NBSIR 85-3179 PARASSD: GENERALIZED REPORTING SUBSYSTEM

---

* PARATRANSIT ADVANCED ROUTING AND SCHEDULING SYSTEM DOCUMENTATION.

Figure 1.1 Schematic View of ARSS



2. A HEURISTIC ROUTING/SCHEDULING DIAL-A-RIDE (RSDAR) ALGORITHM

A heuristic algorithm is presented here to solve the dial-a-ride routing and scheduling problem. The basic approach of the algorithm is to assign patrons to tours in such a way that productivity measures of subtours are maximized while all requirements for service quality constraints are satisfied.

2.1 BASIC CONCEPTS

The heuristic algorithm is based on a set of concepts specifically developed for routing and scheduling dial-a-ride patrons. A familiarity with these concepts, as introduced in the following subsections, is essential to the understanding of RSDAR.

2.1.1 Tour, Subtour, Trip and Base Trip

Some of the basic concepts used in RSDAR are those of tour, subtour, trip and base trip. A tour is defined as a collection of subtours assigned to a vehicle over a specified time interval, the elapsed time from the start to the end of a designated work shift for a vehicle. Each time interval in a tour during which the vehicle has at least one patron on board, (i.e., the vehicle is not empty) is a subtour.

A trip can be defined by its three major components: (1) a patron and his characteristics, (2) his origin and destination locations; and (3) his desired pickup time and/or delivery time.

A base trip is tied to the concept of a base patron. In a subtour, the first patron who boards the vehicle is the base patron and his trip is the base trip. All other patrons in the subtour must share some portion of the ride with the base patron. A patron who can not share some portion of a ride with the base patron can not be included in that subtour.

Tour and subtour are associated with the vehicle, while trip and base trip are associated with the patron.

2.1.2 Service Quality Constraints

The dial-a-ride operating authority establishes its service quality by imposing two independent requirements: (1) patron maximum on board time; and (2) pickup/delivery deviation allowance.

Patron maximum on board time limits the time that a patron may stay on board a vehicle, eliminating routings where ride times become excessive. Maximum on board time is the sum of an allowance for excessive ride time and the trip exclusive ride time (i.e., the minimum time for the vehicle to go from the patron's origin location to his destination location). The allowance for excessive ride time can be established by assigning a percentage of the trip exclusive ride, and superimposing a lower bound and an upper bound on the allowance, or it can simply be decreed as a fixed value.

The pickup/delivery deviation allowance time defines another service quality requirement. This allowance establishes the length of a time window within which the pickup or delivery must be made. The patron specified clock time becomes one end point of the time window as described in the next sections.

2.1.3 Pickup and Delivery Time Windows

Trip requests are classified either as pickup specified or as dropoff specified. If a trip request asks for a pickup time, the trip is considered as pickup specified. Likewise, if a trip request asks for a dropoff time, the trip is considered as dropoff specified. A trip request can be either pickup specified or dropoff- specified, but not both.

Two time windows are created for each trip request: one pickup window and one dropoff window. The pickup window is the time interval between the earliest pickup time (EP) and the latest pickup time (LP). The dropoff window is the time interval between the earliest dropoff time (ED) and the latest dropoff time (LD). When a pickup specified trip request calls for a pickup after a certain time, the specified time becomes the earliest pickup time (EP), and the latest pickup time (LP) is the earliest pickup time plus the deviation allowance time. However, if a pickup-specified trip request calls for a pickup before a certain time the specified time becomes the latest pickup time (LP), and the earliest pickup time is the latest pickup time minus the deviation allowance time. For a pickup-specified trip request, the earliest dropoff time (ED) is the earliest pickup time plus the trip exclusive ride time, and the latest dropoff time (LD) is the latest pickup time plus the maximum on board time.

For dropoff-specified trips, the process is reversed with the dropoff window established first. If a trip calls for a dropoff before a certain time, the specified time becomes the latest dropoff time (LD), and the earliest dropoff time (ED) is the latest dropoff time minus deviation allowance time. On the other hand, if a trip calls for a dropoff after a certain time, the specified time becomes the earliest dropoff time (ED) and the latest dropoff time (LD) is the earliest dropoff time plus the deviation allowance time. The windows for pickup-specified and dropoff-specified trips are illustrated in Figure 2.1.

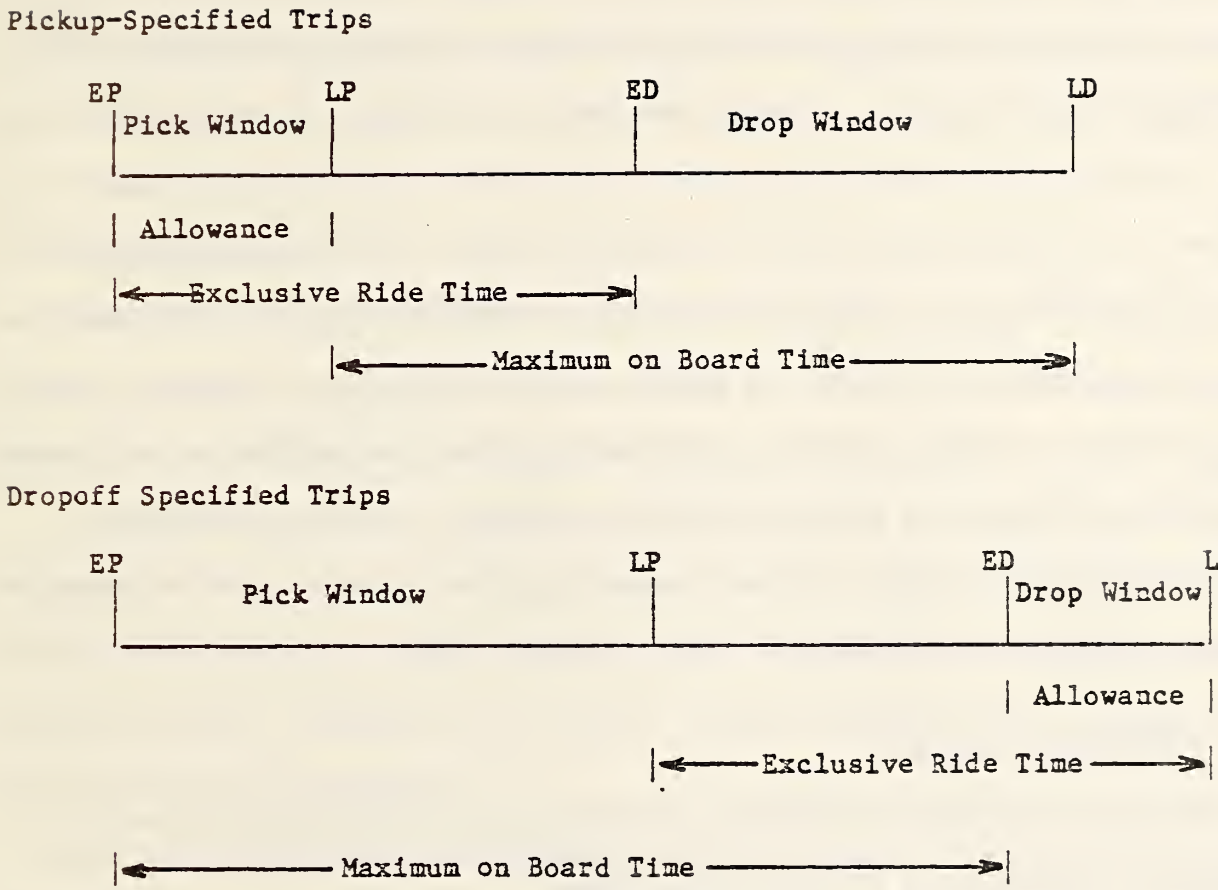


Figure 2.1 Time Windows for Pickup Specified and Dropoff-Specified Trips

A trip's scheduled pickup and delivery times have to fall within the pickup and delivery windows, respectively. However, this does not guarantee that the trip is scheduled feasibly because the windows are so constructed that both pickup and delivery times may be within their respective windows, but the trip time can exceed the maximum on board time. Therefore, the RSDAR algorithm also checks the feasibility of maximum on board time.

2.1.4 Priority

In a dial-a-ride system, each trip is assigned a priority depending upon the patron's characteristics. RSDAR uses the trip priority as assigned in CONENV. Priority (i.e., the relationship with patron characteristics) is determined by the local dial-a-ride management system. The RSDAR algorithm is capable of handling up to fifteen priorities. According to the precedence rules within the assignment procedure, no patron can be assigned to a subtour if a patron of higher priority is yet to be assigned. Thus the priorities are useful not only for ordering, but for stratification as well. The best subtour is selected based on the trips of highest priority. The precedence rules provide a mechanism for implicit enumeration.

2.1.5 Productivity Measures

Productivity measures are used in the algorithm to help achieve the optimization objectives in a dial-a-ride system.

The productivity measures that are currently defined in the RSDAR are:

1. Number of patrons;
2. Number of trips;
3. Number of patrons per unit travel time;
4. Number of trips per unit travel time;
5. Patron exclusive ride times per unit travel time; and
6. Trip exclusive ride time per unit travel time.

All of these measures represent either work or work per unit time. Items (1) through (4) consider the unit of work to be patrons (or trips); all patrons (trips) have equal value. Items (1) and (2) are integer counts; items (3) and (4) are utilization factors. Items (5) and (6) consider the unit of work to be the patron (trip) exclusive ride time; the value of a trip is proportional to its length.

2.1.6 Dwell Time

Dwell time is the time spent loading or unloading patrons to or from a vehicle. Number of patrons, characteristics of patrons, type of vehicle and whether it is loading or unloading, all affect vehicle dwell time at a stop.

A dwell time file is constructed for each type of vehicle. Each file has two major headings, pickup and drop-off dwell time. Under each major heading are three subheadings, namely, non-wheelchair (regular patrons), lift (those patrons who need a vehicle lift to get on board a vehicle), and wheelchair (patrons who use wheelchairs).

Empirical or estimated time values for each category are stored for each number of patrons up to the vehicle capacity. This format is used to accommodate observed non-linearities (e.g., n patrons can board/alight at a single stop in less time than is required for a single patron at each of n stops). In the routing and scheduling, numbers of each type of patron (whether they belong to a single trip or multiple trips) at a vehicle stop are first determined, then the dwell times are obtained for the various types. The largest time value is considered the dwell time of this stop. Thus, with mixed patron types, the largest dwell time is considered the total dwell time.

The pickup dwell time is included in a schedule before the scheduled pickup time; the drop-off dwell time enters a schedule after the scheduled drop-off time. If a number of trips in a subtour have the same stop location, the total dwell time is consolidated into a single entry and is shown as the first scheduled pickup time or as the last scheduled drop-off time depending upon the type of stop.

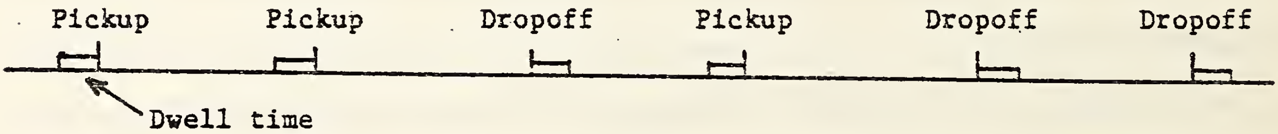


Figure 2.2 The Relative Position of Dwell Time to Scheduled Pickup Time and Dropoff Time

2.1.7 Vehicle Capacity

Vehicle capacity is defined in terms of combinations of patron types. For example, a vehicle may accommodate both wheelchair and non wheelchair patrons with capacity defined as:

with no wheelchairs, twelve non-wheelchair patrons;

with one wheelchair, nine non-wheelchair patrons;

with two wheelchairs, seven non-wheelchair patrons;

with three wheelchairs, three non-wheelchair patrons;

and with four wheelchairs, no non-wheelchair patrons.

2.1.8 Mobility Code

A mobility code with four levels is built in the dial-a-ride system. The four levels are: (1) extended wheelchair; (2) wheelchair; (3) lift; and (4) regular or non-handicapped. Each patron has a single code; for multiple riders on a single trip, the code corresponding to the least mobile rider is used. An extended wheelchair patron is considered less mobile than a wheelchair patron;

a wheelchair patron, less mobile than a lift patron; and a lift patron, less mobile than a regular or non-handicapped one. Vehicles have multiple code classifications which indicate capability.

The RSDAR algorithm checks the compatibility of patron mobility code with the vehicle mobility classification code. If they are not compatible, then the patron can not be assigned to the vehicle.

2.2 HEURISTIC APPROACHES

To facilitate understanding of the heuristic algorithm, it is appropriate to describe some basic concepts of the algorithm before stating the procedures.

2.2.1 Time Interval

To generate a tour, the heuristic algorithm creates subtours in time intervals, terminating when all time intervals have been exhausted or no patrons can be assigned to the tour. The first task is to set in the starting and ending times of a given tour, and any scheduled breaks such as lunch, coffee breaks, etc. The algorithm then picks the first time interval, usually from tour starting time to the starting time of the first break. If no break is scheduled, this time interval ends with the ending time of the tour. The algorithm attempts to form a subtour in the interval. When a subtour is formed, there will be two intervals, one immediately preceding and one immediately succeeding the subtour. The algorithm attempts to create a subtour in the first interval. If no subtour can be formed in an interval, it is considered to have been examined and the algorithm moves to the next interval. A tour is completed after the last interval is examined.

2.2.2 Maximum Remaining Time

One measure used in RSDAR to decide which patron is to be included in a subtour is the remaining time of the base trip. The remaining time of the base trip is obtained, first by including the patron's trip in the subtour, then subtracting base trip travel time from the maximum on board time of the base trip. The underlying rationale is that the larger the remaining time of a base trip, the greater the opportunity to add new patrons to the subtour.

In a given time interval, a base trip (a subtour of two stops) is first tentatively chosen; the algorithm then expands this two-stop subtour by inserting more patrons. The process of expansion adds only one patron at a time, hence the subtour grows from two stops to four stops, then six, and so on. At any stage of expanding a subtour from n stops to n+2 stops, the algorithm examines all combinations of n+2 stops which do not affect the internal order of the n stops for patrons already assigned, selecting the one which allows the largest remaining time for the base patron, which in turn identifies the patron to be added in the subtour. The process of expansion will be repeated until no feasible trip can be added to the subtour. In short, the maximum remaining time of a base trip is the selection criterion for adding an additional patron to a subtour.

2.2.3 Productivity Measure

In a given time interval, subtours are formed based on a maximum remaining time criterion. There are as many subtours as the number of feasible patrons. However, only one subtour, among all possible subtours, can be selected in a time interval. The selection criterion for a subtour is the productivity measure.

In a time interval, the first subtour formed becomes the incumbent; any subsequently formed subtour with higher productivity will replace the incumbent. The process of selecting a subtour in a time interval terminates after all qualified patrons are examined.

Since each trip is assigned a priority and each priority is related to a productivity measure, comparisons of subtours are made by their productivity measures for the highest priority. If there is a tie, then the productivity measures for the next highest priority will be considered.

2.2.4 Base Trip Matrix

The most time-consuming part of the algorithm is the formation of subtours. For each subtour, a base trip is first selected, then the size of the subtour is increased by adding one patron at a time. Every unassigned (unscheduled) patron is examined for possible ride-sharing with the base trip patron, and the patron providing the maximum remaining time of the base trip is selected to be included in the subtour. The process is repeated until no unassigned patrons can be added to the subtour.

To avoid examining every unassigned patron repetitively, the algorithm uses the Base Trip Matrix, constructed in CONENV and updated in RSDAR as tours are constructed, which defines all pairwise ride sharing. In the Base Trip Matrix, each row represents a base patron and each column represents a patron. Elements of the Base Trip Matrix have values of one or zero. A one in an element means that the patron can share a ride with the base patron; a zero means that the patron can not share a ride with the base patron. The Base Trip Matrix effectively reduces the number of patrons to be examined for every base trip.

2.2.5 Compatibility Matrix

The Compatibility Matrix is of the same 0-1 form as the Base Trip Matrix, an element of "1" meaning that the row-column patrons can share a ride (be on the same vehicle at the same time); a "0", that they cannot share a ride. The Compatibility Matrix is a mechanism by which the mixing of classes of patrons can be avoided. Criteria for making this determination are site specific; they could include physical characteristics (elderly, mentally retarded, juvenile, etc.) and policy characteristics (priorities, sponsoring agencies, etc.). These are expected to be so different from one site to another that it is not appropriate to write a general procedure for this purpose.

In some instances, incompatibility can result not from patron characteristics, but from the times and geography of origins and destinations. For example, candidates A and B may each be potential sharing riders for base patron C but not be able to share with each other.

The Compatibility Matrix as constructed in the current version of CONENV uses only geographic considerations. However, RSDAR does not require any interpretation of the share/no share criteria. Therefore, the structure and procedure require no internal modification.

2.2.6 Potential Upper Bound

Another effort to reduce the number of subtours to be built is to set a bounding constraint in the algorithm, the productivity measure of the incumbent subtour. Initially, a potential upper bound is obtained for every patron from the Base Trip Matrix. These potential upper bounds are revised as patrons are assigned to tours. Since a subtour has a productivity measure for each priority, the appropriate productivity measure has to be used as a potential upper bound.

After an incumbent subtour is formed and its productivity measure obtained, new base patrons are examined to form new subtours. Before starting the process of constructing new subtours, the potential upper bounds of new base patrons are compared with the productivity measure of the incumbent subtour. If the potential upper bound of a base patron is smaller than the productivity measure of the incumbent subtour, this base patron is skipped for further investigation in the current time interval. The rationale is as follows: if the potential upper bound of the base patron is smaller than the productivity measure of the incumbent subtour, then this base patron can never form a subtour with a productivity measure higher than that of the incumbent subtour. Therefore, one can skip this base patron.

This procedure is an implicit, rather than total, enumeration. Subtour selection is based on the highest productivity measure of the highest priority. Subsequent insertion of lower priority tours is made in priority order.

2.2.7 Time and Locations of Tour Start, End, and Break

Every stop in a tour has two attributes, time and location. The time and location for stops of tour start, tour end, and tour break are input to RSDAR.

Every tour start time, tour end time and tour break time are entered as time windows. Stops that do not have pre-assigned time windows can simply be assigned a 24-hour time interval. It should be pointed out that the duration of a tour break is a necessary input to RSDAR. In terms of data storage, both a tour start-tour end pair and break start-break end pair are equivalent to dummy trips and appear in the patron list as such.

A location (the x,y coordinate) for each tour stop is generally required in RSDAR. However, in cases of tour start, end, and break, locations are not necessary. If the locations are not given, then the location of the stop succeeding the tour start stop in a tour is considered as the location for the tour start, and the location of the stop preceding the tour break or tour end will be considered as the location for the tour break or tour end.

The ability to have an unspecified location allows for a break in service scheduled with respect to time, but at any location. This is useful for coffee breaks, lunch periods, etc. It also allows for "artificial" tours; these correspond to a hired-by-ride shared-ride taxicab. Artificial tours are treated exactly as regular tours except that each may have only one subtour. This allows for short term (one subtour) utilization of a vehicle which can appear whenever and wherever needed and disappears after the assigned subtour is completed.

2.2.8 Shifting

Since a subtour is formed in a time interval, a larger time interval provides a better opportunity to build a larger subtour. Therefore, it is useful to shift the two ends of a time interval to make the interval as large as possible. In order to shift the two ends of an interval, it is necessary to shift all subtours preceding the interval to the earliest feasible time and all subtours following the interval to the latest feasible time. The algorithm implements the above idea by initially assigning all patrons in the subtour being formed to the latest feasible time. After the subtour is formed, the algorithm will examine the time interval preceding this subtour. If no subtour can be formed in the interval, then the algorithm will shift all scheduled times of the subtour from latest feasible time to earliest feasible time.

2.2.9 Call Back

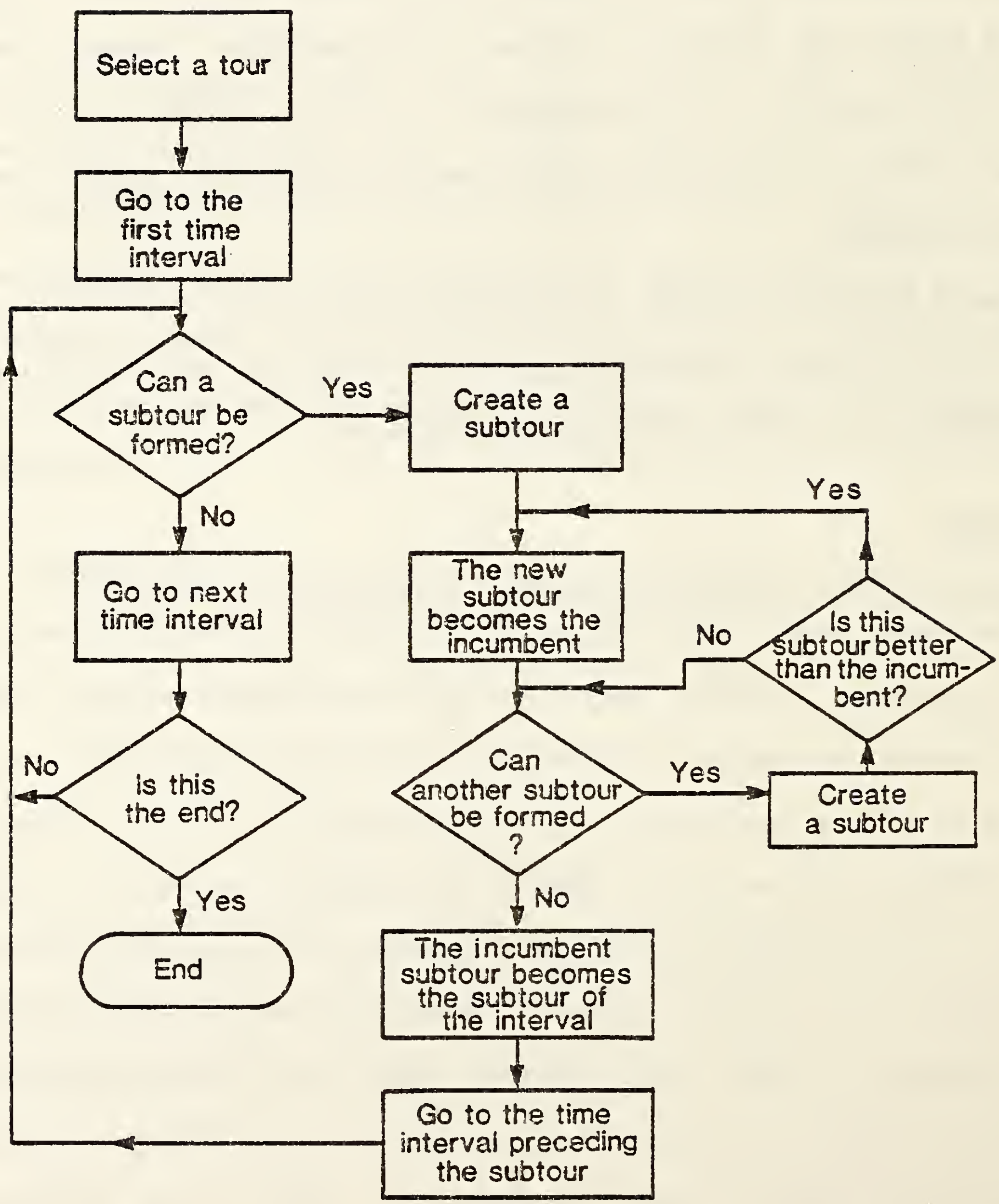
It becomes necessary to call back a patron if his scheduled pickup time is not in his promised pickup time interval. A patron gets a promised pickup time interval when he calls in to make a trip request. However, this promised pickup time interval may not be the pickup time interval that RSDAR uses to schedule the patron. Due to the way that time windows are constructed, when a patron specifies a pickup time, his pickup time interval will coincide with the promised pickup time interval. However, if he specifies a dropoff time, his pickup time interval will not necessarily be the same as the promised pickup time interval. In this case, his scheduled time can be outside the promised time interval.

RSDAR has a subroutine CALBCK to check every patron if a call back is necessary. If so, affected patrons will be marked for call back and, in the report subsystem, a call back list will be generated.

2.3 FLOW CHART

The logical flow of the RSDAR algorithm is depicted in Figure 2.3.

Figure 2.3 Flow Diagram of the RSDAR Algorithm



3. FUNCTIONAL DESCRIPTION OF RSDAR

3.1 THE COMPUTER PROGRAM

The computer program for RSDAR consists of sixteen subroutines. Seven form the main stream of the RSDAR algorithm, the other nine provide various utility functions. A simple flowchart of RSDAR in terms of these seven subroutines is depicted in Figure 3.1; descriptions of all subroutines in RSDAR are provided in Appendix H. The complete listing of RSDAR can be found in Appendix I. However, a more detailed description of the seven mainstream subroutines can be found in the following sections.

3.1.1 Subroutine MAIN

Subroutine MAIN performs the following major functions: It declares values of parameters which are sizes of various array dimensions; specifies external units for input and output files; calls Subroutine READCV to read the input file; calls Subroutine MNTOUR to perform routing and scheduling for all trip requests; calls Subroutine CALBCK to check the need for calling patrons; and, finally, calls Subroutine DUMPGR to write all output results on disk files.

A PARAMETER statement is used in Subroutine MAIN to declare values of eleven parameters which are used as "named constants" for array dimensions throughout RSDAR. Detailed descriptions of these parameters can be found in Section 3.3.

The specification of external units for input and output files is another major function of Subroutine MAIN. The Advanced Routing and Scheduling System has three subsystems, namely, the preprocessor, CONENV, the routing and scheduling algorithm, RSDAR, and the report generator, GREPOR. Since RSDAR is the second subsystem, it takes output of CONENV as its input, and its output is the input to GREPOR. Hence OPEN statements are used to bind file names with unit numbers of external devices.

The naming convention of files used in RSDAR subsystem is compatible with that in CONENV and GREPOR subsystems. All file names are of the form XX.XXX. The first letter is always z, the second letter indicates the subsystem that produces the file. Only three letters are involved: c means CONENV, g means GREPOR, and r means RSDAR. The three letters after the period denote the reading subsystem. For example, zr.gre means the file is produced by RSDAR and is to be read as input by GREPOR.

The input data to be read is in the file 'zc.rsd', and is assigned to external unit 11 with a status of 'old', meaning that the file should exist. The output file is 'zr.gre' and is assigned to external unit 21 with a status of 'new', meaning that the file cannot yet exist, but will be created.

After the input data are read, Subroutine MAIN calls Subroutine MNTOUR to initiate routing and scheduling of all trip requests, returning to Subroutine MAIN when the task of routing and scheduling is finished. At this time, Subroutine MAIN will call Subroutine CALBCK to identify those patrons whose scheduled times are not within their promised times. Patrons on this list must be telephoned to notify them of the change.

Finally, Subroutine MAIN will call Subroutine DUMPCC to write all RSDAR output on file 'zr.gre', so that the report generating subsystem will be able to process these data. Details of the output will be discussed in Section 3.4.

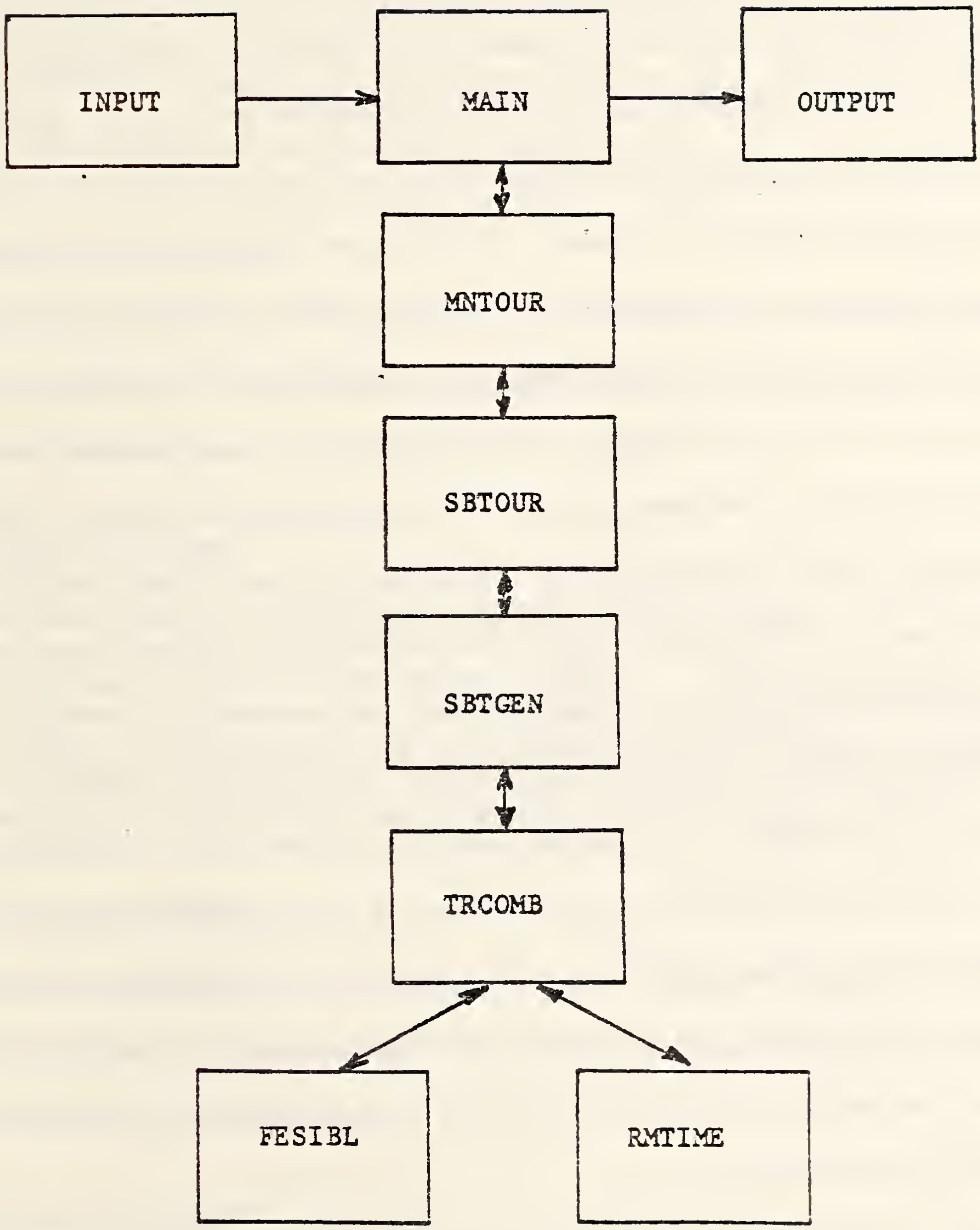


Figure 3.1 Flow Diagram of Subroutines of RSDAR

3.1.2 Subroutine MNTOUR

Subroutine MNTOUR is the center of activities, performing many tasks. It calculates initial upper bounds of productivity measures and determines the minimum size of time interval in which the algorithm will build subtours.

The subroutine searches the patron list and the tour list to see if there are any unassigned patrons or tours. If both are found, the subroutine will enter tour information, determine the time interval within which subtours will be built, initialize the priority sought to the highest, and select an unassigned patron with the highest estimated value of productivity measure for the sought priority as the base patron. It then calls Subroutine SBTOUR to build a subtour with the base patron. When Subroutine SBTOUR returns with a subtour, Subroutine MNTOUR will search for unassigned patrons whose upper bound of productivity measure is higher than the productivity measure of the current subtour and send them, one at a time, to Subroutine SBTOUR to build new subtours. The process terminates when the patron list is exhausted. If no subtour can be built with the base patron of the highest priority within a time interval, Subroutine MAIN then will advance the sought priority to the next highest priority and iterate until either a subtour is built in the time interval or the priority list is exhausted, indicating that no subtour can be built in the time interval.

A tour is completed when the last time interval in the tour has been examined.

Subroutine MNTOUR returns the control to Subroutine MAIN when either all patrons have been assigned to a tour or all available tours have been used.

3.1.3 Subroutine SBTOUR

The major functions of Subroutine SBTOUR are to set the stage to call Subroutine SBTGEN for expansion of the subtour for the given base patron, and to keep the best subtour that has been established.

3.1.4 Subroutine SBTGEN

Subroutine SBTGEN screens patrons for ridesharing with the base patrons; then it calls Subroutine TRCOMB to conduct further investigation. Patrons are added to the subtour one at a time.

3.1.5 Subroutine TRCOMB

Subroutine TRCOMB systematically inserts the two stops for the new patron into the existing order of stops of the subtour. It then calls Subroutine FESIBL to check the feasibility of the specific order of patron stops, and Subroutine RMTIME to calculate the remaining time of the base trip.

3.1.6 Subroutine FESIBL

Subroutine FESIBL checks whether the given order of patron stops of the subtour is feasible in terms of service quality criteria, such as time windows and maximum on board time.

3.1.7 Subroutine RMTIME

Subroutine RMTIME calculates the remaining free time from the maximum on-board time of the base patron for the currently entered tentative stop sequence of the subtour, saving the stop sequence with the larger remaining time.

The complete listing of RSDAR can be found in Appendix I.

3.2 INPUT DATA REQUIREMENTS

The RSDAR Subsystem assumes the existence of many data files to be read as input. These data files are a subset of files generated by the CONENV subsystem.

The input data to be read by RSDAR are a set of variables and a set of arrays. The variables to be read in are PA03VX,PA03XX,PA07XX,PA10XX,PA08XX, PA08RX,IBX and IDX. These variables are defined below:

PA03VX: Number of Vehicle Entries in Candidate Trip List (one/tour + one/break).

PA03XX: Total Number of Entries in Candidate Trip List (Vehicles + Trips).

PA07XX: Number of Vehicle Types.

PA08RX: Number of Real Tours.

PA08XX: Number of Available Tours (Real + Artificial).

PA10XX: Number of Entries in Vehicle Capacity-Dwell List.

IBX: Local Index Variable for Base Trip Matrix.

IDX: Local Index Variable for Distance Matrix.

Nine arrays (namely, VTYPEL, VCADWE, TOURLI, CANDTP, BASETP, COMPAT, DISTOO, DISTOD, and PRIORL) are also read in as input. They are described in the following paragraphs.

The VTYPEL array contains, for each type of vehicle, mobility and priority types that the vehicle can accommodate, the maximum capacity for non-wheelchair patrons, characteristics of vehicle, and a pointer to array VCADWE.

The VCADWE array contains information on vehicle dwell time due to patrons boarding or debarking the vehicle. VCADWE has three patron categories, non-wheelchair, lift, and wheelchair, and two types of dwell time, pick dwell and drop dwell for each. Dwell times are indexed by the number of non-wheelchair patrons.

The VCADWE array also provides the vehicle capacity for wheelchair patrons. All combined capacities of wheelchair patrons and non-wheelchair patrons can also be found there.

Dwell times of vehicle types are stored consecutively in VCADWE. For each type, a pointer in VTYPEL indicates the corresponding first row of dwell times.

The TOURLI array lists information on vehicles assigned to tours. For every available tour, this array stores the vehicle index, type, and patron priorities that the vehicle can handle.

The CANDTP array is the candidate trip list; it contains information on a trip required by a patron and stops required of the vehicle. For each trip request, the CANDTP uses sixteen words to store all relevant information.

The BASETP array is a matrix used to indicate permissible ride sharing of base trips. The BASETP rows represent base trips, columns of the matrix represent sharing trips, and elements of the matrix are either zero or one. If $a_{ij} = 1$, trip j can share a ride with base trip i; if $a_{ij} = 0$, trip j can not share a ride with base trip i.

BASETP also presents the number of trips, by priority, that can share a ride with each base trip.

The COMPAT array is similar to the BASETP array: its elements, b_{ij}, indicate whether trip i and trip j can be assigned to the same subtour.

The DISTOO and DISTOD arrays contain distance (or equivalent travel time) matrices. The DISTOO array stores distances from origin to origin or destination to destination. The origin-to-origin distances are stored in the upper right part of the matrix, and the destination-to-destination distances in the lower left part of the array. They are separated by the diagonal elements, which are all zero's.

The DISTOD array contains the distance matrix of origin to destination. A row of the matrix represents origin index, and a column of the matrix represents destination index.

Distances are actually stored in terms of travel time (to the nearest 1/8 of a minute) at a nominal speed. Since only ten bits of storage are allocated for this element, travel distance is limited to a maximum of 128 minutes at nominal speed. To save memory space, three distance elements are stored in one word in DISTOO and DISTOD.

The PRIORL array contains the assigned productivity code for each priority, a productivity measure calculated in subroutine PRDCTY. PRIORL also stores numbers of patrons in each priority.

3.3 PARAMETERS

The PARAMETER statement in Subroutine MAIN declares values of eleven parameters which are used as "named constants" for array dimensions throughout RSDAR.

Since RSDAR may be installed in different computer systems and may have to handle extremely diverse dial-a-ride situations, array dimensions in all subroutines are expected to be changed to satisfy the need. Using parameter declaration makes it easier to locate and change these values. Changes are limited to Subroutine MAIN; other subroutines need not be recompiled.

1. PARM01. It dimensions arrays BASETP and COMPAT.
The minimum value should be ((PA03PX/31)+4)*PA03PX, where PA03PX is the number of trip requests. All divisions performed in estimating values of parameters are rounded up to the next higher integer.
The divisor is 31 because the first bit of a 32-bit word is a sign bit. In BASETP and COMPAT, information is stored in a single bit.

2. PARM02: It dimensions array BVECT1, and the minimum value should be (PA03PX/31)+4.

3. PARM03: It dimensions array CANDTP and the minimum value is set to be ICAX, the maximum number of entries for CANDTP. The entries for CANDTP are of three types: tour start, tour break, and patron trip. Each tour start, tour break, and patron trip needs an entry in CANDTP.

4. PARM04: It dimensions arrays DISTOD and DISTOO and the minimum value is $(PA03XX)^2/3$.

5. PARM05: It dimensions arrays IBSBTR, ICSBTR, ISE, and KSE. Since it is the maximum number of stops in a subtour, the value is estimated by an "educated guess."

6. PARM06: It dimensions array IVWHTR. Since it is the maximum number of stops in a tour, the value is estimated by an "educated guess."

7. PARM07: It dimensions array VTYPEL. The minimum value is the number of vehicle types.

8. PARM08: It dimensions array TOURLI. The minimum value is the number of tours.

9. PARM09: It dimensions array TSTOPL. The minimum value is 2*PA03XX.

10. PARM10: It dimensions array VCADWE. The minimum value is $\sum_i (1+C_i)$ where C_i is the non-wheelchair capacity of vehicle type i.

11. PARM11: It dimensions array PRDBD and the minimum value is PA03PX.

3.4 MODEL OUTPUT

The outputs of the RSDAR model are stored in three arrays: two of them, CANDTP and TOURLI appear in input and are augmented by RSDAR. The third file, TSTOPL, the tour stop list, is created in RSDAR.

The following items are associated with each patron in the CANDTP array as part of the output of RSDAR.

1. Scheduled* and expected pickup time;
2. Scheduled and expected dropoff time;
3. Tour assignment;
4. Pickup stop in the tour;
5. Dropoff stop in the tour;
6. Callback indicator; and
7. Locations for tour stops if necessary

Two items are to be added to every tour in the TOURLI array: the number of stops assigned to the tour, and the number of the row in TSTOPL where tour stops begin.

The TSTOPL array contains the tour stop list, storing the following items for every stop of each tour.

(1) Type of stop.

(A) pickup

(B) dropoff

(C) break start

(D) break end

(E) tour start

(F) tour end

---

* In RSDAR, scheduled and expected times have the same value. The expected time is included here to accommodate monitoring and <u>ex post facto</u> reporting functions.

(2) Dwell time. During the vehicle dwell time (expressed in 1/8ths of a minute) at a stop, patrons board or disembark the vehicle.

(3) Patron index. This is the candidate trip list, making accessible all information regarding the patron in CANDTP.

(4) Number of non-wheelchair patrons on board for next leg.

(5) Number of lift patrons on board for next leg.

(6) Number of wheelchair patrons on board for next leg.

(7) Travel time. The time, expressed in minutes, to travel from this stop to the next.

(8) Slack time. This is the time the vehicle is idle, expressed in minutes.

(9) Possible delay time. The difference between latest drop time and scheduled drop time shows the possible delay time for this stop without violating its time window.

4. OPERATIONAL CHARACTERISTICS OF RSDAR

4.1 COMPUTATIONAL RESULTS WITH RSDAR

A sample problem, collected from a one-day paratransit operation, was used for all test runs. There were 300 patron trip requests, 33 of which were for wheelchair patrons. The data file first listed all wheelchair trip requests, then the regular trip requests. Within each patron type, the data were chronologically ordered. The trip request data included patron's name, required pickup time, type of patron, number in the party, and locations (X,Y coordinates) of trip origin and destination. Appendix J lists the complete data file for 300 patron trip requests.

A set of external conditions influences the routing and scheduling of patrons. As set in CONENV, these are read as input to the RSDAR subsystem, including information on:

1. service quality criteria;
2. productivity measures for each priority;
3. patron characteristics;
4. dwell time;
5. vehicle characteristics;
6. number of tours available;
7. tour starting, break and ending times and locations; and
8. names of patron sponsoring agency.

Two series of test runs were made on the development system (see section 4.3 for description). Each series had seven runs, the first run started with 25 patrons, and the number of patrons increased by 25 for each succeeding run. The last run had 175 patrons. The two series of test runs differed in the productivity measure used in selecting subtours. The first series of test runs used number of patrons as the productivity measure for all patrons in the run. The second series used number of patrons as the productivity measure for wheelchair patrons and the ratio of number of patrons per unit time as the productivity measure for regular patrons. The details of run times, number of tours, and number of patrons assigned for the two series of test run are listed in Table 4.1 and Table 4.2 respectively; the numbers of stops in each for every test run is listed in Table 4.3 and Table 4.4. Wheelchair patrons were assigned a priority of 1; others were assigned a priority of 15.

From these runs, it was observed that the heuristic algorithm is very sensitive to input data. The run time appears to vary approximately cubically with respect to the number of patrons. That the the run time varies with the productivity measure used can be seen in Table 4.1 and Table 4.3 for the two series of test runs.

To explore the effect of the number of priorities on the computer running time, eight test runs were carried out. Using 100 non-wheelchair patrons from the data base, each run was given a number of priorities, then these 100 patrons were uniformly distributed over the number of priorities. The test results are in Table 4.5.

Table 4.1 Results of #1 Test Series (Using Productivity Measure #1 for All Patrons)

Number of Patrons	Run Time	Number of Tours	Number of Patrons Assigned
25	1:17	7	25
50	5:11	10	50
75	16:07	20	75
100	1:02:38	25	100
125	1:34:20	29	125
150	2:13:47	32	149
175	2:48:46	32	172

Table 4.2 Results of #2 Test Series (Using Productivity Measure #1 for Wheelchair Patrons and Productivity Measure #3 for Regular Patrons)

Number of Patrons	Run Time	Number of Tours	Number of Patrons Assigned
25	1:12	7	25
50	6:25	11	50
75	29:46	22	75
100	1:28:26	27	100
125	2:33:12	30	125
150	3:29:37	32	146
175	5:03:12	32	170

Table 4.3 Tour Assignments of #1 Test Series

Tour Number	Number of Patrons						
	25	50	75	100	125	150	175
1*	4	7	7	8	8	9	10
2*	6	9	7	7	7	10	8
3	4	5	5	6	6	6	6
4	4	6	9	8	8	7	9
5	4	5	7	7	8	9	8
6	2	4	3	5	5	5	5
7	1	4	2	2	7	3	6
8		4	4	5	3	5	6
9		3	4	4	4	4	9
10		3	5	4	6	8	8
11			3	4	5	6	8
12			3	4	4	8	7
13			3	4	4	4	6
14			3	3	4	5	7
15			2	3	4	5	6
16			2	3	6	4	6
17			2	3	4	6	8
18			2	3	4	4	4
19			1	3	3	3	4
20			1	3	3	3	4
21				3	3	3	5
22				3	3	4	3
23				3	3	3	3
24				1	3	3	5
25				1	2	4	3
26					2	4	3
27					2	3	3
28					2	3	3
29					2	2	2
30						2	3
31						2	2
32						2	2
Number of Tours	7	10	20	25	29	32	32
Number of Patrons Assigned	25	50	75	100	125	149	172

*The first and second tours include a mandatory break stop.

Table 4.4 Tour Assignments of #2 Test Series

Tour Number	Number of Patrons						
	25	50	75	100	125	150	175
1*	4	7	7	8	8	9	10
2*	6	7	7	7	7	8	8
3	4	5	5	6	9	8	8
4	4	6	6	6	5	6	6
5	4	7	6	6	6	8	8
6	2	4	5	5	7	5	7
7	1	4	4	4	5	6	6
8		4	4	5	3	5	6
9		3	4	4	6	8	8
10		1	4	3	4	6	8
11		2	4	4	6	5	8
12			2	14	4	3	6
13			2	14	5	6	5
14			2	3	6	4	7
15			1	2	5	4	5
16			2	3	4	5	5
17			3	3	3	5	5
18			2	3	4	4	4
19			2	3	2	3	4
20			1	2	3	5	4
21			1	3	3	3	3
22			1	3	3	3	5
23				2	2	3	4
24				3	3	3	3
25				2	3	5	6
26				1	2	2	6
27				1	2	3	2
28					2	2	3
29					1	3	3
30					1	2	2
31						1	3
32						3	2
Number of Tours	7	11	22	27	30	32	32
Number of Patrons Assigned	25	50	75	100	125	146	170

*The first and second tours include a mandatory break stop.

Number of Priorities	Run Time	Number of Tours	Number of Subtours
1	2:47:28	26	34
2	1:23:20	29	40
3	1:08:48	29	37
4	1:10:18	30	40
6	59:14	31	41
8	58:45	32	41
10	1:04:36	30	39
15	1:10:13	30	38

Table 4.5 Test Results of Different Number of Priorities

Test runs were also made by perturbing the length of the time interval for patron starting times. It was observed that both the computer run time and the number of tours decreased monotonically as the length of the time interval increased. For a set of 100 patrons, the starting times of all patrons were distributed uniformly within a time interval of 100 minutes, the RSDAR generated a schedule of 29 tours and the computer run time was about one hour and thirty six minutes. When the time interval was increased to 400 minutes, the schedule had only 14 tours and the run time was only about twenty minutes.

4.2 COMPUTER SYSTEM CONFIGURATION[1]

All computer runs described in this report were executed on a WICAT System 150. The processor for the WICAT System 150 is the Motorola 68000 which runs at 8 MHz and executes approximately one million instructions per second. This processor has a 16-bit external data path, but internally it has

[1]Identification of commercial products is included only to adequately specify the test procedure. Identification does not imply recommendation or endorsement by the National Bureau of Standards.

32-bit registers and supports 32-bit operations. The computer system hardware configuration has a memory size of 512 KB of dynamic RAM, a 15 MB 5 1/4" Winchester disk drive, a 960KB 5 1/4" floppy disk drive, a cathode ray terminal, and a 132 character/line hard copy printer.

The operating system for the WICAT-150 is a WICAT version of UNIX. A FORTRAN 77 compiler is also a part of the system software.

4.3 COMPUTER HARDWARE AND SOFTWARE REQUIREMENTS

The minimum requirements for computer hardware to be able to run the RSDAR subsystem are specified as follows:

1. The Central Processing Unit (CPU) requires 32-bit internal registers to support 32-bit data operations. Lengths greater than 32-bits are acceptable.
2. The memory size requires at least 512 kilobytes.
3. The mass storage requires a Winchester disk drive of at least 15 megabyte capacity and at least one floppy disk drive for backup and file handling purposes.
4. A CRT (24 line by 80 character) which generates and displays the 96 character ASCII set.
5. A high-quality printer which has a line width of 132 characters and handles tractor feed pages.

The computer software requirement for RSDAR are a FORTRAN complier which implements FORTRAN-77 and an operating system with file handling and editing capability.

Appendix A INTERNAL DATA FORMATS

This appendix defines all global data arrays used in RSDAR. The specifications are given in a series of tables.[1] The tables appear in alphabetic order of variable name. Each specification includes:[2]

1 - Array name (e.g., CANDTP)

2 - Dimensioning parameter (e.g., PARM03)

3 - Counts and subcounts as necessary (e.g., PA03VX, PA03RX, PA03XX)

4 - Definition (e.g., Deviation Both [i.e., patron and vehicle])

5 - Element length: bits or characters (e.g., 7 bits)

6 - Bit/word/field relations (e.g., bits 2-8, word 7, Deviation)

7 - Units (e.g., minutes)

8 - Range or discrete allowable values (e.g., 0-127)

9 - Coding or interpretation (e.g., not used)

---

[1]The tables are numbered consecutively for all global arrays for the system (CONENV, RSDAR, and GREPOR). Only those tables (A.10 through A.39) for arrays in RSDAR appear in this appendix.

[2]Examples given in the text are based on an entry from table A.12.

Table A.10 - BASETP BASE TriP matrix
PARM01,PA01XX

Name	Word	1st Bit	Length	Units	Range	Coding
Priority 1 Trips	1	2	7	-	0-127	
Priority 2 Trips		9	8	-	0-255	
Priority 3 Trips		17	8	-	0-255	
Priority 4 Trips		25	8	-	0-255	
Priority 5 Trips	2	2	7	-	0-127	
Priority 6 Trips		9	8	-	0-255	
Priority 7 Trips		17	8	-	0-255	
Priority 8 Trips		25	8	-	0-255	
Priority 9 Trips	3	2	7	-	0-127	
Priority 10 Trips		9	8	-	0-255	
Priority 11 Trips		17	8	-	0-255	
Priority 12 Trips		25	8	-	0-255	
Priority 13 Trips	4	2	7	-	0-127	
Priority 14 Trips		9	8	-	0-255	
Priority 15 Trips		17	8	-	0-255	
All Trips		25	8	-	0-255	
Incidence Bits	5	2-32	1	-	0-1	By bit for trips 1-31 0→not potential sharer 1→potential sharer

Words 6, 7, etc. are analagous to word 5 for trips 32-62, 63-93 etc, up to total number of patrons.

There are as many replications of the sequence as necessary for all patron base trips.

Table A.11 - BVECT1: Base trip matrix row VECTor

Name	Word	1st Bit	Length	Units	Range	Coding
Priority 1 Trips	1	2	7	-	0-127	
Priority 2 Trips		9	8	-	0-255	
Priority 3 Trips		17	8	-	0-255	
Priority 4 Trips		25	8	-	0-255	
Priority 5 Trips	2	2	7	-	0-127	
Priority 6 Trips		9	8	-	0-255	
Priority 7 Trips		17	8	-	0-255	
Priority 8 Trips		25	8	-	0-255	
Priority 9 Trips	3	2	7	-	0-127	
Priority 10 Trips		9	8	-	0-255	
Priority 11 Trips		17	8	-	0-255	
Priority 12 Trips		25	8	-	0-255	
Priority 13 Trips	4	2	7	-	0-127	
Priority 14 Trips		9	8	-	0-255	
Priority 15 Trips		17	8	-	0-255	
All Trips		25	8	-	0-255	
Incidence Bits	5	2-32	1	-	0-1	By bit for trips 1-31 0→not potential sharer 1→potential sharer

Words 6, 7, etc. are analagous to word 5 for trips 32-62, 63-93 etc, up to total number of patrons.

Table A.12 - CANDTP CANDidate TrIPs
PARM03,PA03VX,PA03RX,PA03XX

Name	P/V	Word	1st Bit	Length	Units	Range	Coding
Stop Type	Both	1	2	2	-	0-2	0→Patron 1→Break 2→End
Trip Type	Both		4	2	-	0-3	0→Vehicle 1→Advanced 2→Immediate 3→Deferred
Trip Status	Both		6	5	-	0-1	By Bit
Assigned	-		6	1	-	0-1	1→Assigned
In Process	-		7	1	-	0-1	1→In process
Completed	-		8	1	-	0-1	1→Completed
Cancelled	-		9	1	-	0-1	1→Cancelled
No Show	-		10	1	-	0-1	1→No Show
Mobility Codes	Both		11	3	-	0-1	By Bit
Extended WC			11	1	-	0-1	1→Extended WC
Wheelchair			12	1	-	0-1	1→Wheelchair
Lift			13	1	-	0-1	1→Lift
Patron Codes	Patron		14	3	-	0-1	By Bit
Call Back			14	1	-	0-1	1→Call Back
Pick/Drop Specification			15	1	-	0-1	1→Drop
Before/After Specification			16	1	-	0-1	1→After
Pick Stop Index	Both		17	8	-	0-255	Within Tour
Drop Stop Index			25	8	-	0-255	Within Tour
Report Flag	Both	2	2	3	-	0-7	≠0→Selected
Priority	Patron		5	4	-	1-15	

Table A.12 - CANDTP CANDidate TrIPs (continued)

Name	P/V	Word	1st Bit	Length	Units	Range	Coding
Vehicle Type	Vehicle		5	4	-	1-15	
Characteristics	Both		9	8	-	0-1	By Bit 1→Existence of Characteristic
Agency	Both		17	8	-	1-127	
Vehicle Index	Both		25	8	-	0-255	
Tour		3	2	7	-	0-127	
Exclusive Ride Time	Patron		9	12	1/8 min.	0-511 7/8	
Break Duration	Vehicle		9	12	1/8 min.		
Max on Board Time	Patron		21	12	1/8 min.	0-511 7/8	
Expected Pick Time	Both	4	2	15	1/8 min.	0-4095 7/8	
Expected Drop Time	Both		17	16	1/8 min.	0-8191 7/8	
Pick X	Both	5	2	15	1/8 min.	0-4095 7/8	
Drop X	Both		17	16	1/8 min.	0-8191 7/8	
Pick Y	Both	6	2	15	1/8 min.	0-4095 7/8	
Drop Y	Both		17	16	1/8 min.	0-8191 7/8	
Deviation	Both	7	2	7	minutes	0-127	Not Used
Pick Zone	Both		9	8	-	1-255	

Table A.12 - CANDTP CANDidate TriPs (continued)

Name	P/V	Word	1st Bit	Length	Units	Range	Coding
Drop Zone	Both	7	17	8	-	1-255	
Pick Code	Both		25	4	-	0-15	
Drop Code	Both		29	4	-	0-15	
Identification	Patron	8	2	31	-		
Priority Capability	Vehicle		18	15	-	0-1	By Bit 1→Capability
Phone Number	Patron	9	2	31	-		
Earliest Pick Time	Both	10	2	31	1/8 min.		
Scheduled Pick Time	Both	11	2	31	1/8 min.		
Latest Pick Time	Both	12	2	31	1/8 min.		
Earliest Drop Time	Both	13	2	31	1/8 min.	-	
Scheduled Drop Time	Both	14	2	31	1/8 min.		
Latest Drop Time	Both	15	2	31	1/8 min.		
Non Handicapped Patrons	Patron	16	5	4	-	0-15	
Lift Patrons	Patron		9	4	-	0-15	
Wheelchair Patrons	Patron		13	4	-	0-15	
Pick Dwell Time	Both		17	8	1/8 min.	0-31 7/8	
Drop Dwell Time	Both		25	8	1/8 min.	0-31 7/8	

Table A.13 - COMPAT COMPATibility matrix
PARM01,PA01XX

Name	Word	1st Bit	Length	Units	Range	Coding
Priority 1 Riders	1	2	7	-	0-127	
Priority 2 Riders		9	8	-	0-255	
Priority 3 Riders		17	8	-	0-255	
Priority 4 Riders		25	8	-	0-255	
Priority 5 Riders	2	2	7	-	0-127	
Priority 6 Riders		9	8	-	0-255	
Priority 7 Riders		17	8	-	0-255	
Priority 8 Riders		25	8	-	0-255	
Priority 9 Riders	3	2	7	-	0-127	
Priority 10 Riders		9	8	-	0-255	
Priority 11 Riders		17	8	-	0-255	
Priority 12 Riders		25	8	-	0-255	
Priority 13 Riders	4	2	7	-	0-127	
Priority 14 Riders		9	8	-	0-255	
Priority 15 Riders		17	8	-	0-255	
All Riders		25	8	-	0-255	
Incidence Bits	5	2-32	1	-	0-1	By bit for trips 1-31 0→not potential sharer 1→potential sharer

Words 6, 7, etc. are analogous to word 5 for trips 32-62, 63-93 etc, up to total number of patrons.

There are as many replications of the sequence as necessary for all patron base trips.

Table A.14 - DISTOD DISTance Origin-Destination
PARM04,PA04XX

Name	Word	1st Bit	Length	Units	Range	Coding
Distance Pick 1 to Drop 1	1	3	10	1/8 min.	0-127 7/8	
Pick 1 to Drop 2		13	10			
Pick 1 to Drop 3		23	10			

DISTOD is a square matrix stored as a vector.

Table A.15 - DISTOO DISTance Origin-Origin
PARM04,PA04XX

Name	Word	1st Bit	Length	Units	Range	Coding
Distance Pick 1 to Pick 1	1	3	10	1/8 min.	0-127 7/8	
Pick 1 to Pick 2		13	10			
Pick 1 to Pick 3		23	10			

DISTOO is a square matrix stored as a vector. The upper triangular portion has pick-to-pick times; the lower triangular portion, drop-to-drop times.

Table A.17 - IBSBTR BeSt SuBTouR
PARM05

Name	Word	1st Bit	Length	Units	Range	Coding
Patron index	1	2	31			
Stop type	2	2	31			
NWC on board	3	2	31			
Lift on board	4	2	31			
WC on board	5	2	31			
Travel time (next leg)	6	2	31	1/8 min.		
Dwell time	7	2	31	1/8 min.		
Scheduled time	8	2	31	1/8 min.		
Number of NWC	9	2	31			
Number of Lift	10	2	31			
Number of WC	11	2	31			
Patron priority	12	2	31			

Table A.18 - ICSBTR Current SuBTouR
PARM05

Name	Word	1st Bit	Length	Units	Range	Coding
Patron index	1	2	31			
Stop type	2	2	31			
NWC on board	3	2	31			
Lift on board	4	2	31			
WC on board	5	2	31			
Travel time (next leg)	6	2	31	1/8 min.		
Dwell time	7	2	31	1/8 min.		
Scheduled time	8	2	31	1/8 min.		
Number of NWC	9	2	31			
Number of lift	10	2	31			
Number of WC	11	2	31			
Patron priority	12	2	31			

Table A.20 - ISE Working Area of Subtour PARM05

Name	Word	1st Bit	Length	Units	Range	Coding
Patron index	1	2	31			
Stop type	2	2	31			
NWC on board	3	2	31			
Lift on board	4	2	31			
WC on board	5	2	31			
Travel time (next leg)	6	2	31	1/8 min.		
Dwell time	7	2	31	1/8 min.		
Scheduled time	8	2	31	1/8 min.		
Number of NWC	9	2	31			
Number of Lift	10	2	31			
Number of WC	11	2	31			
Patron priority	12	2	31			

Table A.21 - IVWHTR WHole TouR
PARM06

Name	Word	1st Bit	Length	Units	Range	Coding
Patron index	1	2	31			
Stop type	2	2	31			
NWC on board	3	2	31			
Lift on board	4	2	31			
WC on board	5	2	31			
Travel time (next leg)	6	2	31	1/8 min.		
Dwell time	7	2	31	1/8 min.		
Scheduled time	8	2	31	1/8 min.		
Idle time	9	2	31	1/8 min.		
Potential delay time	10	2	31	1/8 min.		

Table A.22 - KSE Storage Area of Subtour PARM05

Name	Word	1st Bit	Length	Units	Range	Coding
Patron index	1	2	31			
Stop type	2	2	31			
NWC on board	3	2	31			
Lift on board	4	2	31			
WC on board	5	2	31			
Travel time (next leg)	6	2	31	1/8 min.		
Dwell time	7	2	31	1/8 min.		
Scheduled time	8	2	31	1/8 min.		
Number of NWC	9	2	31			
Number of Lift	10	2	31			
Number of WC	11	2	31			
Patron priority	12	2	31			

Table A.25 - PRDBD PRoDuctivity bounDs
PARM11

Name	Word	1st Bit	Length	Units	Range	Coding
Productivity bound in Priority #1 for patron 1	1	2	31			
Productivity bound in Priority #2 for patron 1	2	2	31			
Productivity bound in Priority #3 for patron 1	3	2	31			
Productivity bound in Priority #4 for patron 1	4	2	31			
Productivity bound in Priority #5 for patron 1	5	2	31			
Productivity bound in Priority #6 for patron 1	6	2	31			
Productivity bound in Priority #7 for patron 1	7	2	31			
Productivity bound in Priority #8 for patron 1	8	2	31			
Productivity bound in Priority #9 for patron 1	9	2	31			
Productivity bound in Priority #10 for patron 1	10	2	31			
Productivity bound in Priority #11 for patron 1	11	2	31			
Productivity bound in Priority #12 for patron 1	12	2	31			
Productivity bound in Priority #13 for patron 1	13	2	31			
Productivity bound in Priority #14 for patron 1	14	2	31			
Productivity bound in Priority #15 for patron 1	15	2	31			

There are as many replications of the sequence as necessary for all patrons.

Table A.26 - PRDCOD PRoDuctivity CODE of priority 15,15

Name	Word	1st Bits	Length	Units	Range	Coding
Productivity code of priority #1	1	2	31			
Productivity code of priority #2	2	2	31			
Productivity code of priority #3	3	2	31			
Productivity code of priority #4	4	2	31			
Productivity code of priority #5	5	2	31			
Productivity code of priority #6	6	2	31			
Productivity code of priority #7	7	2	31			
Productivity code of priority #8	8	2	31			
Productivity code of priority #9	9	2	31			
Productivity code of priority #10	10	2	31			
Productivity code of priority #11	11	2	31			
Productivity code of priority #12	12	2	31			
Productivity code of priority #13	13	2	31			
Productivity code of priority #14	14	2	31			
Productivity code of priority #15	15	2	31			

Table A.27 - PRDIB PRoDuctIvities of Best subtour
15,15

Name	Word	1st Bits	Length	Units	Range	Coding
Productivity of priority #1 for current base patron	1	2	31			
Productivity of priority #2 for current base patron	2	2	31			
Productivity of priority #3 for current base patron	3	2	31			
Productivity of priority #4 for current base patron	4	2	31			
Productivity of priority #5 for current base patron	5	2	31			
Productivity of priority #6 for current base patron	6	2	31			
Productivity of priority #7 for current base patron	7	2	31			
Productivity of priority #8 for current base patron	8	2	31			
Productivity of priority #9 for current base patron	9	2	31			
Productivity of priority #10 for current base patron	10	2	31			
Productivity of priority #11 for current base patron	11	2	31			
Productivity of priority #12 for current base patron	12	2	31			
Productivity of priority #13 for current base patron	13	2	31			
Productivity of priority #14 for current base patron	14	2	31			
Productivity of priority #15 for current base patron	15	2	31			

Table A.28 - PRDIC PRoDuctIvities of Current subtour
15,15

Name	Word	1st Bits	Length	Units	Range	Coding
Productivity of priority #1 for current base patron	1	2	31			
Productivity of priority #2 for current base patron	2	2	31			
Productivity of priority #3 for current base patron	3	2	31			
Productivity of priority #4 for current base patron	4	2	31			
Productivity of priority #5 for current base patron	5	2	31			
Productivity of priority #6 for current base patron	6	2	31			
Productivity of priority #7 for current base patron	7	2	31			
Productivity of priority #8 for current base patron	8	2	31			
Productivity of priority #9 for current base patron	9	2	31			
Productivity of priority #10 for current base patron	10	2	31			
Productivity of priority #11 for current base patron	11	2	31			
Productivity of priority #12 for current base patron	12	2	31			
Productivity of priority #13 for current base patron	13	2	31			
Productivity of priority #14 for current base patron	14	2	31			
Productivity of priority #15 for current base patron	15	2	31			

Table A.29 - PRIORL PRIORity List
15,15

Name	Word	1st Bit	Length	Units	Range	Coding
Priority	Index	-	-	-	1-15	
Number Advanced	1	2	11	-	0-2047	
Number Immediate		12	12	-	0-4095	
Productivity Code		29	4	-	1-15	

Table A.30 - PRIOTR Priority TRuth table
15,15

Name	Word	1st Bits	Length	Units	Range	Coding
Is priority #1 accepted by current tour ?	1	2	31			
Is priority #2 accepted by current tour ?	2	2	31			
Is priority #3 accepted by current tour ?	3	2	31			
Is priority #4 accepted by current tour ?	4	2	31			
Is priority #5 accepted by current tour ?	5	2	31			
Is priority #6 accepted by current tour ?	6	2	31			
Is priority #7 accepted by current tour ?	7	2	31			
Is priority #8 accepted by current tour ?	8	2	31			
Is priority #9 accepted by current tour ?	9	2	31			
Is priority #10 accepted by current tour ?	10	2	31			
Is priority #11 accepted by current tour ?	11	2	31			
Is priority #12 accepted by current tour ?	12	2	31			
Is priority #13 accepted by current tour ?	13	2	31			
Is priority #14 accepted by current tour ?	14	2	31			
Is priority #15 accepted by current tour ?	15	2	31			

1 - indicates yes and 0 indicates no.

Table A.35 - TOURLI TOUR LIst
PARM08,PA08RX,PA08UX,PA8XX

Name	Word	1st Bit	Length	Units	Range	Coding
Tour Number	Index					
Vehicle Index	1	2	7	-	1-127	
Vehicle Type		9	4	-	1-15	
Priority Capability		18	15	-	0-1	By Bit: 1 → ability to accommodate indexed priority.
Report Flag	2	2	3	-	0-7	Used in reporting only
Number of Stops		5	12	-	0-4095	
TSTOPL Pointer		17	16	-	0-64K	0th entry in TSTOPL
Agency	3	2	7	-	0-127	
Characteristics Capability		9	8	-	0-1	By Bit: 1→ability to accommodate indexed characteristic

Table A.37 - TSTOPL Tour STOP List
PARM09,PA09XX

Name	Word	1st Bit	Length	Units	Range	Coding
Stop Type	1	2	3	-	1-6	1→Pick 2→Drop 3→Start Break 4→End Break 5→End Tour 6→Start Tour
Dwell Time		5	8	1/8 min.	0-31 7/8	
Candidate Index		13	16	-	$1-2^{15}$	
Report Flag		29	4	-	0-1	
On Board-Next Leg	2					
NHC Riders		2	7	-	0-127	
Lift Riders		9	8	-	0-255	
Wheelchair Riders		17	8	-	0-255	
Times-Next Leg	3					
Travel		2	7	minutes	0-127	
Slack		9	12	minutes	0-4K	
Maximum Offset Time		21	12	minutes	0-4K	
Report Pointers	4					
Candidate Pointer		16	15	-	0-32K	
Report Item		31	2	-	0-2	0→Candidate 1→Drop Stop 2→Pick Stop

Table A.38 - VCADWE Vehicle CApacity and DWEll
PARM10,PA10XX

Name	Word	1st Bit	Length	Units	Range	Coding
Record 1						
WC Capacity	1	2	7	-	0-15	WC capacity with 0 NHC
	2	-	-	-	--	Not Used
Record I (I=2,NHC Capacity+1)						
WC Capacity	1	2	7	-	0-15	With I-1 NHC
NHC Pick Dwell Time		9	8	1/8 min.	0-31 7/8	For I-1 riders
Lift Pick Dwell Time		17	8	1/8 min.	0-31 7/8	For I-1 riders
WC Pick Dwell Time		25	8	1/8 min.	0-31 7/8	For I-1 riders
NHC Drop Dwell Time	2	9	8	1/8 min.	0-31 7/8	For I-1 riders
Lift Drop Dwell Time		17	8	1/8 min.	0-31 7/8	For I-1 riders
WC Drop Dwell Time		25	8	1/8 min.	0-31 7/8	For I-1 riders

Table A.39 - VTYPEL Vehicle TYPE List
PARM07,PA07XX

Name	Word	1st Bit	Length	Units	Range	Coding
Vehicle Type	Index	-	-	-	1-15	
Mobility Code		2	3	-	0-7	First three characteristics
		2	1	-	0-1	1→'X' capability
		3	1	-	0-1	1→'W' capability
		4	1	-	0-1	1→'L' capability
Non Handicapped Capacity		5	4	-	1-15	
Characteristic Capability		9	8	-	0-1	By Bit 1→capability indexed
VCADWE Pointer		17	16	-	0-64K	To 0th VCADWE for this vehicle type

APPENDIX B INPUT FORMATS

Appendix B, within this series of reports, is reserved for input formats.

Since RSDAR uses no input, Appendix B is empty.

APPENDIX C · INTERFACE FORMATS

This appendix defines the use, content, and format of the files which are interfaces to RSDAR. Content is as extracted from a FORTRAN READ/WRITE statement. The format specification is as extracted from a FORMAT statement. To avoid confusion between the number 0 and the letter O representing an octal FORMAT, a lower case o will be used.

1. File zc.rsd

This file is created by CONENV for input to RSDAR. Its content is those global variables required for execution of RSDAR plus several variables used for setting read specifications. The content, format and example reference are as follows:

```
1 - Outer loop - none
    Content - PA01XX, PA02XX, PA03XX, PA04XX, PA05XX, PA06XX, PA07XX,
              PA08XX, PA09XX, PA10XX, PA11XX, PA12XX, PA13XX, PA14XX,
              PA15XX, PA16XX, PA17XX, PA18XX, PA03VX, PA08RX, IBX, IDX.
              IBX is the number of words/base trip
              IDX = PA04XX
    Format -  (20o6)

2 - Outer loop - none
    Content - (VTYPEL [I], I=1, PA07XX)
    Format - (8o12)

3 - Outer loop - (I=1, PA10XX)
    Content - (I,VCADWE [I,J], J=1,2)
    Format - (I5, 2o12)

4 - Outer loop - (I=1, PA08XX)
    Content - (I, TOURLI [I,J], J=1,3)
    Format - (I5,3o12)

5 - Outer loop - (I=1, PA03XX)
    Content - (I,CANDTP [I,J], J=1,16)
    Format - (I5,8o12/5X,8o12)

6 - Outer loop - (I=1, PA03XX-PA03VX)
    Content - (I,BASETP [I,J], J=1, IBX)
              (I,COMPAT [I,J], J=1, IBX)
    Format - (I5,8o12/5X,8o12) for each write

7 - Outer loop - (I=1, IDX)
    Content - (I,DISTOO [I], DISTOD [I])
    Format - (I5,2o12)

8 - Outer loop - (I=1,15)
    Content - (I, PRIORL [I])
    Format - (I5,o12)
```

2. File zr.gre

This file is created by RSDAR for input to GREPOR. Its content is some global variables created by RSDAR and some from zc.rsd whose values have been modified by RSDAR. The content, format, and example reference are as follows:

1 - Outer loop - none
Content - PA09XX, PA08UX
Format - (20I2)

2 - Outer loop - (I=1, PA03XX)
Content - (CANDTP [I,J], J=1,16)
Format - (80I2)

3 - Outer loop - (I=1, PA09XX)
Content - (TSTOPL [I,J], J=1,4)
Format - (80I2)

4 - Outer loop - (I=1, PA08UX)
Content - (TOURLI [I,J], J=1,3)
Format (80I2)

APPENDIX D OUTPUT FORMATS

Appendix D, within this set of reports, is reserved for output formats. Since RSDAR produces no output, this appendix is empty.

APPENDIX E SAMPLE INPUT

Appendix E, within this series of reports, is reserved for a sample input set. The sample input is never referenced in the RSDAR exposition; therefore this appendix is empty.

APPENDIX F SAMPEL INTERFACES

Appendix F, in this series of reports is reserved for listings of interface files produced by execution of CONENV and RSDAR with the sample input of Appendix E. Since the sample input is not referenced in the RSDAR exposition, this appendix is empty.

APPENDIX G SAMPLE OUTPUT

Appendix G, in this series of reports, is reserved for a sample output set.

There is no ouput from RSDAR; this appendix is empty.

APPENDIX H SUBROUTINE SPECIFICATIONS

This appendix provides a brief description of all subroutines in the model. For the MAIN program, a more detailed description by functional block is given in Chapter 3. The appendix is arranged in alphabetical order by subroutine name. Each subroutine is described on a single program summary sheet. This summary sheet includes: (a) the name of the subroutine; (b) the mnemonic of the subroutine; (c) the purpose of the subroutine; (d) the called routines; (e) the arguments used; (f) the calling routines; (g) input from file storage; (h) output to file storage. The information provided on these summary sheets in conjunction with the model flowchart shown in Chapter 3 should facilitate the programmer's task of effectively maintaining the model. The interactions among subroutines which are explicitly stated on the summary sheets should also assist the programmer in making any modifications to the source code dictated by user needs or peculiarities of the operating system.

Subroutine	CALBCK - CALl BaCK
Purpose	Mark those patron who need a call back because the scheduled pickup time is outside the promised time window.
Called by	MAIN
Arguments	CANDTP PA03VX PA03XX
Calls	FIELD

Subroutine	DUMPGR - DUMP to GRepor
Purpose	Write output of RSDAR on a disk file, zr.gre.
Called by	MAIN
Arguments	PA03XX PA08UX PA09XX CANDTP TOURLI TSTOPL PARM03 PARM08 PARM09
Calls	None
Output	zr.gre interface to GREPOR

Subroutine	DWELLT - DWELL Time
Purpose	Obtain appropriate dwell time from VCADWE array
Called by	TRCOMB
Arguments	VCADWE PARM10
	I1: Number of non-wheelchair patrons to pickup I2: Number of lift patrons to pickup I3: Number of wheelchair patrons to pickup I4: Number of non-wheelchair patrons to drop off I5: Number of lift patrons to drop off I6: Number of wheelchair patrons to drop off
	J1: Starting row in VTYPEL
	DWT: Dwell Time
Calls	FIELD

Subroutine	FESIBL - FEaSIBiLity check
Purpose	Check if the subtour is feasible
Called by	TRCOMB
Arguments	CANDTP,DISTOD,DISTOO,PARM03,PA03XX,PARM04,PARM05,PARM06
	IK2: Number of stops in subtour.
	IFLAG1: =1 : subtour is feasible. =0 : subtour is not feasible.
	IFLAG5: =1 : subtour cut short due to end of interval =0 : otherwise.
	IVWHTR: Whole tour of a vehicle.
	KSE: Storage area of subtour.
	IBGNP: Patron at the beginning of time interval.
	IBGNS: Stop at the beginning of time interval.
	IENDP: Patron at the end of time interval.
	IENDS: Stop at the end of time interval.
Calls	None

Subroutine	FIELD - store or fetch FIELD
Purpose	Pack or unpack bits information into or from a word.
Called by	CALBCK DWELLT MNTOUR PREPRD RMTIME SBTGEN SBTOUR TFETCH TRCOMB
Arguments	I: I=1 means unpacking I=2 means packing
	J: Packed argument
	K: Unpacked argument
	I1: First bit position in packed argument
	I2: Number of bits
Calls	None

Subroutine	MAIN
Purpose	Initialize the run. Call READCV to read input tile; MNTOUR to create tours, and DUMPGR to write output file.
Called by	None
Arguments	None
Calls	READCC MNTOUR CALBCK DUMPCC
Parameters	PARM01,PARM02,PARM03,PARM04,PARM05,PARM06,PARM07,PARM08, PARM09,PARM10,PARM11

Subroutine	MNTOUR - MaiN TOUR
Purpose	1. Calculate productivity upper bounds 2. Read in starting, ending, and break time of a tour, then call SBTOUR to generate subtours for the tour. 3. Repeat step 2 until either all patrons are assigned or all available tours are exhausted.
Called by	MAIN
Arguments	BASETP,CANDTP,COMPAT,DISTOD,DISTOO,PA03PX, PA03VX,PA03XX,PA08R,PRIORL, TOURLI,TSTOPL,VCADWE,VTYPEL

BVECT1: A row vector from BASETP

IBSBTR: Best subtour in existence

ICSBTR: Current subtour

IVWHTR: Whole tour of a vehicle

ISE: Working area of subtour

KSE: Storage area of subtour

PRIOTR: Priority truth table

PRDIB: Productivities of the best subtour

PRDIC: Productivities of the current subtour

PRDBD: Productivity bounds

PRIOSM: Priority sum

PRDCOD: Productivity code of priority

Calls FIELD PRDCTY SHIFT TFETCH SBTOUR

Subroutine	PRDCTY - PRoDuCtiviTY calculation
Purpose	Calculate value of productivity
Called by	PREPRD MNTOUR
Arguments	IPCODE: Productivity code
	PATRN: Number of patrons
	TRIP: Number of trips
	PATRT: Sum of patron ride time
	TRPRT: Sum of trip ride time
	TIME: Total elapsed time of the subtour
	PRDV: Productivity value
Calls	None

Subroutine	PREPRD - PREpare for PRoDuctivity
Purpose	Prepare for calculating value of productivity
Called by	SBTOUR
Arguments	CANDTP,PRDCOD,PARM03,PARM05
	IP: Priority
	ICSBTR: Current subtour
	PRDIC: Productivity of current subtour
	ISCTP: Number of stops in th current subtour
Calls	PRDCTY FIELD

Subroutine	READCV - READ ConenV interface
Purpose	Read input data from disk file
Called by	MAIN
Arguments	BASETP,CANDTP,COMPAT,DISTOD,DISTOO,PRIORL,TOURLI, VCADWE,VTIPEL,PARM03,PARM08,PARM10
	PA03PX: Number of all trip enties in CANDTP PA03VX: Number of all vehicle entries in CANDTP PA03XX: Number of all entries in CANDTP PA08RX: Number of real tour entries in TOURLI PA08XX: Number of all tour entries in TOURLI
Calls	None
Input	zc.rsd interface from CONENV

Subroutine	RMTIME - ReMaining TIME
Purpose	Calculate the remaining free time of base patron
Called by	TRCOMB
Arguments	CANDTP,IK2,ISE,KSE,PARM03,PARM05

IA:	New patron to be inserted
JK:	Number of stops in the subtour
IFLAG2:	=1: A better subtour of IK2 stops is found, =0 otherwise
IC1:	Base patron
IC2:	Scratch
RTB:	Remaining time of the best subtour
RTS:	Remaining tme of subtour
SHTB:	Shortest travel time of the best subtour
SHTS:	Shortest travel time of subtour

Calls	FIELD

Subroutine	SBTGEN - SuBTour GENerator
Purpose	Generate subtours
Called by	SBTOUR
Arguments	IFLAG2 IFLAG5 BVECT1 CANDTP COMPAT DISTOD DISTOO ICSBTR ISE IVWHTR KSE VCADWE IBGNP IBGNS IC1 PA03XX ICSTP IENDP IENDS PA03PX PA03VX PA08UX PARM03 PARM04 PARM05 PARM06 PARM10
	JA: Base patron
	IP1: Priority
	JRDWT: VCADWE pointer
	MCVC: Vehicle Mobility Code
	NWCAP: Non-wheelchair capacity
Calls	FIELD TRCOMB

Subroutine	SBTOUR - SuBTOUR for base patron
Purpose	Obtain the best subtour for the given base patron
Called by	MNTOUR
Arguments	IFLAG5 BVECT1 CANDTP COMPAT DISTOD DISTOO IBSBTR ICSBTR ISE IVWHTR KSE PRDCOD PRIOTR VCADWE PRDIB PRDIC IBGNP IBGNS PA03XX IENDP IENDS PA03PX PA03VX JRDWT MCVC PA08UX NWCAP PARM03 PARM04 PARM05 PARM06 PARM10

JA:	Scratch
TALSB:	Arrival time to subtour
IFLAG3:	=1 : a new subtour is made, =0 : otherwise
BIPR:	Scratch
IPRIOP:	Patron priority
IPRIOT:	Tour priority
IBSTP:	Number of stops in the best subtour.

Calls	FIELD SBTGEN PREPRD

Subroutine	SHIFT
Purpose	Shift a subtour to the earliest feasible time
Called by	MNTOUR
Arguments	CANDTP,IVWHTR,IBGNS,PARM03,PARM06
	NSTPTR: Number of stops in the tour.
Calls	None

Subroutine	TFETCH - Time FETCH
Purpose	Obtain travel time between two stops
Called by	MNTOUR TRCOMB
Arguments	I: Travel time
	J: Candidate row
	K: Candidate column
	IV: DISTOD or DISTOO
Calls	FIELD

Subroutine	TRCOMB - TouR COMBination
Purpose	Generate orders of patron stops to form a subtour
Called by	SBTGEN
Arguments	IFLAG2 IFLAG5 CANDTP DISTOD DISTOO ISE ICSBTR
	IA: New patron to be inserted
	JK: Number of stops in the subtour
Calls	FIELD TFETCH DWELLT FESIBL RMTIME

APPENDIX I PROGRAM LISTINGS

This appendix contains program listings of all subroutines used in RSDAR.

```
      SUBROUTINE CALBCK (CANDTP,PA03VX,PA03XX,PARM03)
      IMPLICIT INTEGER (B-Y)
      DIMENSION CANDTP(PARM03,1)
      PSHIFT=24
      PGAP=240
      IVAX1=PA03VX+1
      DO 200 I1=IVAX1,PA03XX
      CALL FIELD(1,CANDTP(I1,1),I2,15,1)
      IF (I2.EQ.0) GO TO 200
      PTL=CANDTP(I1,I2)-PSHIFT
      PTE=PTL-PGAP
      IF (CANDTP(I1,11).GT.PTL) GO TO 100
      IF (CANDTP(I1,11).LT.PTE) GO TO 100
      GO TO 200
  100 CALL FIELD(2,CANDTP(I1,1),1,14,1)
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DUMPGR(PA03XX,PA08UX,PA09XX,CANDTP,TOURLI,
     1  TSTOPL,PARM03,PARM08,PARM09)
      IMPLICIT INTEGER (B-Y)
      DIMENSION CANDTP(PARM03,1),TOURLI(PARM08,1),TSTOPL(PARM09,1)
      NAG=-1
      WRITE(21,100) PA09XX,PA08UX
  100 FORMAT(80I2)
      DO 150 I=1,PA03XX
      WRITE(21,100) (CANDTP(I,J),J=1,16)
  150 CONTINUE
      WRITE(21,100) NAG,(CANDTP(1,J),J=2,16)
      DO 250 I=1,PA09XX
      WRITE(21,100) (TSTOPL(I,J),J=1,4)
  250 CONTINUE
      WRITE(21,100) NAG,(TSTOPL(1,J),J=2,4)
      DO 350 I=1,PA08UX
      WRITE(21,100) (TOURLI(I,J),J=1,3)
  350 CONTINUE
      WRITE(21,100) NAG,(TOURLI(1,J),J=2,3)
      RETURN
      END
```

```
      SUBROUTINE DWELLT(I1,I2,I3,I4,I5,I6,J1,DWT,VCADWE,PARM10)
      IMPLICIT INTEGER (B-Y)
      DIMENSION VCADWE(PARM10,1)
C     I1:NUMBER OF NWC TO PICK
C     I2:NUMBER OF LIFT TO PICK
C     I3:NUMBER OF WC TO PICK
C     I4:NUMBER OF NWC TO DROP
C     I5:NUMBER OF LIST TO DROP
C     I6:NUMBER OF WC TO DROP
C     J1:STARTING ROW (FROM VTYPEL)
C     DWT:DWELL TIME
      DWT=0
      TP1=0
      TD1=0
      IF (I3.EQ.0) GO TO 100
      J2=J1+I3
      CALL FIELD(1,VCADWE(J2,1),TP2,25,8)
      IF (TP2.GT.TP1) TP1=TP2
  100 IF (I2.EQ.0) GO TO 200
      J2=J1+I2
      CALL FIELD(1,VCADWE(J2,1),TP2,17,8)
      IF (TP2.GT.TP1) TP1=TP2
  200 IF (I1.EQ.0) GO TO 300
      J2=J1+I1
      CALL FIELD(1,VCADWE(J2,1),TP2,9,8)
      IF (TP2.GT.TP1) TP1=TP2
  300 DWT=TP1
      IF (TP1.GT.0) RETURN
      IF (I6.EQ.0) GO TO 400
      J2=J1+I6
      CALL FIELD(1,VCADWE(J2,2),TD2,25,8)
      IF (TD2.GT.TD1) TD1=TD2
  400 IF (I5.EQ.0) GO TO 500
      J2=J1+I5
      CALL FIELD(1,VCADWE(J2,2),TD2,17,8)
      IF (TD2.GT.TD1) TD1=TD2
  500 IF (I4.EQ.0) GO TO 600
      J2=J1+I4
      CALL FIELD(1,VCADWE(J2,2),TD2,9,8)
      IF (TD2.GT.TD1) TD1=TD2
  600 DWT=TD1
      RETURN
      END
```

```
      SUBROUTINE FESIBL(IK2,IFLAG1,IFLAG5,
     1 CANDTP,DISTOD,DISTOO,IVWHTR,KSE,
     2 IBGNP,IBGNS,PA03XX,IENDP,IENDS,
     3 PARM03,PARM04,PARM05,PARM06 )
      IMPLICIT INTEGER (B-Y)
      DIMENSION CANDTP(PARM03,1),DISTOD(1),DISTOO(1)
      DIMENSION IVWHTR(PARM06,1),KSE(PARM05,1)
C     CHECK ROUTING IN KSE FOR FEASIBLE SCHEDULE
C     IFLAG1=1:ROUTING IS FEASIBLE; =0:NOT FEASIBLE
      IFLAG1=0
      IFLAG5=0
      DIFLA=0
      DIFLB=0
      DIFUA=0
      DIFUB=0
      PDWEL=0
      DDWEL=0
C     SEARCH FOR LARGEST DIFLB AND DIFUB
      DO 100 I1=1,IK2
      J1=KSE(I1,1)
      J2=KSE(I1,2)
      X1=KSE(I1,8)
      IF (J2.NE.1) GO TO 20
      X4=CANDTP(J1,10)
      X5=CANDTP(J1,12)
      IF (I1.NE.1) GO TO 40
      CALL TFETCH(TPD,PA03XX,J1,IBGNP,DISTOD)
      PDWEL=KSE(I1,7)
      X6=IVWHTR(IBGNS,8)+IVWHTR(IBGNS,7)+TPD+PDWEL
      IF (X6.GT.X4) X4=X6
      GO TO 40
   20 CONTINUE
      X4=CANDTP(J1,13)
      X5=CANDTP(J1,15)
      IF (I1.NE.IK2) GO TO 40
      CALL TFETCH(TPD,PA03XX,IENDP,J1,DISTOD)
      DDWEL=KSE(I1,7)
      X6=IVWHTR(IENDS,8)-IVWHTR(IENDS,7)-TPD-DDWEL
      IF (X6.LT.X5) X5=X6
   40 CONTINUE
      IF (X1.GE.X4) GO TO 60
      DIFLA=X4-X1
      IF (DIFLA.GT.0.AND.DIFUB.LT.0) RETURN
      IF (DIFLA.GT.DIFLB) DIFLB=DIFLA
      GO TO 100
   60 IF (X1.LE.X5) GO TO 100
      DIFUA=X5-X1
      IF (DIFLB.GT.0.AND.DIFUA.LT.0) RETURN
      IF (DIFUA.LT.DIFUB) DIFUB=DIFUA
  100 CONTINUE
      IF (DIFLB.EQ.0) GO TO 200
C     SHIFT TO THE RIGHT
      DO 150 I1=1,IK2
      J1=KSE(I1,1)
      J2=KSE(I1,2)
      KSE(I1,11)=KSE(I1,8)+DIFLB
```

```
      IF (J2.NE.1) GO TO 120
      IF (KSE(I1,8).LT.CANDTP(J1,10)) RETURN
      IF (KSE(I1,8).GT.CANDTP(J1,12)) RETURN
      GO TO 140
  120 CONTINUE
      IF (KSE(I1,8).LT.CANDTP(J1,13)) RETURN
      IF (KSE(I1,8).GT.CANDTP(J1,15)) RETURN
  140 CONTINUE
      IF (I1.NE.IK2) GO TO 150
      IF (J2.EQ.1) GO TO 150
      CALL TFETCH(TPD,PA03XX,IENDP,J1,DISTOD)
      X6=IVWHTR(IENDS,8)-IVWHTR(IENDS,7)-TPD-DDWEL
      IF (KSE(I1,8).GT.X6) GO TO 400
  150 CONTINUE
      GO TO 300
  200 IF (DIFUB.EQ.0) GO TO 300
C     SHIFT TO THE LEFT
      DO 250 I1=1,IK2
      J1=KSE(I1,1)
      J2=KSE(I1,2)
      KSE(I1,8)=KSE(I1,8)+DIFUB
      IF (J2.NE.1) GO TO 220
      IF (KSE(I1,8).LT.CANDTP(J1,10)) RETURN
      IF (KSE(I1,8).GT.CANDTP(J1,12)) RETURN
      GO TO 240
  220 CONTINUE
      IF (KSE(I1,8).LT.CANDTP(J1,13)) RETURN
      IF (KSE(I1,8).GT.CANDTP(J1,15)) RETURN
  240 CONTINUE
      IF (I1.NE.1) GO TO 250
      IF (J2.EQ.2) GO TO 250
      CALL TFETCH(TPD,PA03XX,J1,IBGNP,DISTOD)
      X6=IVWHTR(IBGNS,8)+IVWHTR(IBGNS,7)+TPD+PDWEL
      IF (KSE(I1,8).LT.X6) RETURN
  250 CONTINUE
  300 CONTINUE
C     NOT TO EXCEED THE MAXIMUM ON BOARD TIME
      DO 350 I1=1,IK2
      TD=0
      IF (KSE(I1,2).EQ.2) GO TO 350
      J1=KSE(I1,1)
      I2=I1+1
      DO 340 I3=I2,IK2
      IF (KSE(I3,1).EQ.J1.AND.KSE(I3,2).EQ.2) J3=I3
  340 CONTINUE
      TD=KSE(J3,8)-KSE(I1,8)
      CALL FIELD(1,CANDTP(J1,3),TMXOB,21,12)
      IF (TD.GT.TMXOB) RETURN
  350 CONTINUE
      IFLAG1=1
      RETURN
  400 CONTINUE
C     IFLAG5=1 : SUBTOUR CUT SHORT DUE TO END INTERVAL
C     IFLAG5=0 : OTHERWISE
      IFLAG5=1
      RETURN
```

```
      END
```

```
      SUBROUTINE FIELD(I,J,K,I1,I2)
      DIMENSION NV(32)
*    POWERS OF 2 TABLE
      DATA NV /1,2,4,8,16,32,64,128,256,512,1024,2048,4096,8192,
     * 16384,32768,65536,131072,262144,524288,1048576,2097152,
     * 4194304,8388608,16777216,33554432,67108864,134217728,
     * 268435456,536870912,1073741824,2147483648/
*
      I3=34-I1-I2
      J1=J/NV(I3)
      I4=I2+1
      IF (I.EQ.2) GO TO 100
*
*    UNPACKING
      K=J1
      IF (I4.GE.32) RETURN
      J2=J1/NV(I4)
      K=J1-J2*NV(I4)
      RETURN
*
*    PACKING
  100 J2=0
      J3=0
      IF (J4.GE.32) GO TO 110
      J3=NV(I4)
      J2=J1/J3
  110 J=J-J1*NV(I3)+(J2*J3+K)*NV(I3)
      RETURN
      END
```

```
C       PROGRAM RSDAR
C
C       MAXIMUM CAPACITY: PA03XX <= 249; PA03PX <= 217
C
C       CALCULATION OF ARRAY SIZES
C       (NOTE: ALL NUMBERS SHOULD BE ROUNDED TO NEXT HIGHER INTEGER)
C       PARM01: ((PA03PX/31)+4)*PA03PX: BASETP,COMPAT
C       PARM02: (PA03PX/31)+4: BVECT1
C       PARM03: PA03XX: CANDTP: MAX NUMBER OF ENTRIES FOR CANDTP
C       PARM04: (PA03XX**2)/3: DISTOD,DISTOO
C       PARM05: GUESS: IBSBTR,ICSBTR,ISE,KSE: MAX NUMBER OF STOPS
C               IN A SUBTOUR
C       PARM06: GUESS: IVWHTR: MAX NUMBER OF STOPS IN A TOUR
C       PARM07: INFORMATION OR GUESS: VTYPEL
C       PARM08: INFORMATION OR GUESS: TOURLI: MAX NUMBER OF TOURS
C       PARM09: 2*PA03XX: TSTOPL
C       PARM10: INFORMATION OR GUESS: VCADWE
C       PARM11: PRDBD
C
        IMPLICIT INTEGER (B-Y)
        PARAMETER( PARM01=2387, PARM02=11, PARM03=249, PARM04=20667,
     1 PARM05=40, PARM06=80, PARM07=15, PARM08=50, PARM09=498,
     2 PARM10=50, PARM11=217)
        DIMENSION BASETP(PARM01),BVECT1(PARM02),CANDTP(PARM03,16)
        DIMENSION COMPAT(PARM01),DISTOD(PARM04),DISTOO(PARM04)
        DIMENSION IBSBTR(PARM05,12),ICSBTR(PARM05,12),ISE(PARM05,12)
        DIMENSION IVWHTR(PARM06,10),KSE(PARM05,12),PRDCOD(15)
        DIMENSION PRIORL(15),PRIOTR(15),TOURLI(PARM08,3)
        DIMENSION TSTOPL(PARM09,4),VCADWE(PARM10,2),VTYPEL(PARM07)
        DIMENSION PRDIB(15),PRDIC(15),PRDBD(PARM11,15)
        OPEN(11,FILE='zc.rsd',STATUS='OLD')
        OPEN(21,FILE='zr.gre',STATUS='NEW')
        CALL READCV(PA03XX,PA03PX,PA08XX,PA08RX,PA03VX,BASETP,CANDTP,
     1 COMPAT,DISTOD,DISTOO,PRIORL,TOURLI,VCADWE,VTYPEL,
     2 PARM03,PARM08,PARM10)
C
        CALL MNTOUR(BASETP,BVECT1,CANDTP,COMPAT,DISTOD,DISTOO,
     1 IBSBTR,PA03XX,ICSBTR,PA03PX,ISE,PA08RX,PA03VX,IVWHTR,
     2 KSE,PRDCOD,PRIORL,PRIOTR,TOURLI,TSTOPL,VCADWE,VTYPEL,
     3 PRDIB,PRDIC,PRDBD,PA08UX,PA09XX,
     4 PARM03,PARM04,PARM05,PARM06,PARM08,
     5 PARM09,PARM10,PARM11 )
        CALL CALBCK(CANDTP,PA03VX,PA03XX,PARM03)
        CALL DUMPGR(PA03XX,PA08UX,PA09XX,CANDTP,TOURLI,
     1  TSTOPL,PARM03,PARM08,PARM09)
        CLOSE(11)
        CLOSE(21)
        STOP
        END
```

```
      SUBROUTINE MNTOUR(BASETP,BVECT1,CANDTP,COMPAT,DISTOD,DISTOO,
     1 IBSBTR,PA03XX,ICSBTR,PA03PX,ISE,PA08RX,PA03VX,IVWHTR,
     2 KSE,PRDCOD,PRIORL,PRIOTR,TOURL1,TSTOPL,VCADWE,VTYPEL,
     3 PRDIB,PRDIC,PRDBD,PA08UX,PA09XX,
     4 PARM03,PARM04,PARM05,PARM06,PARM08,
     5 PARM09,PARM10,PARM11 )
      IMPLICIT INTEGER (B-Y)
      DIMENSION BASETP(1),BVECT1(1),CANDTP(PARM03,1),COMPAT(1)
      DIMENSION DISTOD(1),DISTOO(1),IBSBTR(PARM05,1),ICSBTR(PARM05,1)
      DIMENSION ISE(PARM05,1),IVWHTR(PARM06,1),KSE(PARM05,1),PRDCOD(1)
      DIMENSION PRIORL(1),PRIOTR(1),TOURL1(PARM08,1),TSTOPL(PARM09,1)
      DIMENSION VCADWE(PARM10,1),VTYPEL(1),PRDIB(1),PRDIC(1)
      DIMENSION PRDBD(PARM11,1),PRIOSM(15)
      DO 130 I1=1,15
      CALL FIELD(1,PRIORL(I1),PRIOSM(I1),1,12)
      CALL FIELD(1,PRIORL(I1),PRDCOD(I1),29,4)
  130 CONTINUE
      PA08UX=0
      PA09XX=0
      IVAX1=PA03VX+1
      IVTYP=0
C     ** CALCULATE UPPER BOUNDS **
      DO 170 J4=1,15
      IF (PRIOSM(J4).EQ.0) GO TO 160
      DO 155 J5=IVAX1,PA03XX
      I1=J5-PA03VX
      I4=PA03PX/31
      I5=PA03PX-31*I4
      IF (I5.NE.0) I4=I4+1
      I4=I4+4
      J1=J4/4
      J2=J4-4*J1
      IF (J2.EQ.0) J2=4
      IF (J2.NE.4) J1=J1+1
      J2=(J2-1)*8+1
      J3=I4*(I1-1)+J1
      CALL FIELD(1,BASETP(J3),TRIP,J2,8)
      IF (TRIP.EQ.0) GO TO 150
      CALL FIELD(1,COMPAT(J3),PATRN,J2,8)
      CALL FIELD(1,CANDTP(J5,3),TIME,9,12)
C
      PATRT=PATRN*TIME
      TRPRT=TRIP*TIME
      CALL PRDCTY(PRDCOD(J4),PATRN,TRIP,PATRT,TRPRT,TIME,PRDV)
      PRDBD(I1,J4)=PRDV
      GO TO 155
  150 PRDBD(I1,J4)=0
  155 CONTINUE
      GO TO 170
  160 DO 165 J3=1,PA03PX
      PRDBD(J3,J4)=0
  165 CONTINUE
  170 CONTINUE
      DO 190 I=IVAX1,PA03XX
      CANDTP(I,4)=0
  190 CONTINUE
```

```
C       ARE ALL PATRONS ASSIGNED ?
  200 CONTINUE
      DO 210 I1=IVAX1,PA03XX
      CALL FIELD(1,CANDTP(I1,1),I2,6,1)
      IF (I2.EQ.0) GO TO 220
  210 CONTINUE
      RETURN
C     **NEW TOUR**
  220 CONTINUE
      GPSIZE=99000
      DO 240 I1=IVAX1,PA03XX
      CALL FIELD(1,CANDTP(I1,1),I2,6,1)
      IF (I2.GT.0) GO TO 240
      CALL FIELD(1,CANDTP(I1,3),I2,9,12)
      IF (GPSIZE.GT.I2) GPSIZE=I2
  240 CONTINUE
      GPSIZE=GPSIZE+80
      DO 255 I1=1,PARM06
      IVWHTR(I1,9)=0
  255 CONTINUE
      ISUBTR=0
      NSTPTR=1
      IBGNS=1
      IVTYPA=IVTYP
      PA08UX=PA08UX+1
      ITOUR=PA08UX
      IF (PA08UX.LE.PA08RX) GO TO 260
      ISUBTR=1
  260 CONTINUE
C     **PRIOTR(I): PRIORITY ACCEPTED BY CURRENT TOUR**
      DO 270 I1=1,15
      I2=I1+17
      CALL FIELD(1,TOURLI(ITOUR,1),PRIOTR(I1),I2,1)
  270 CONTINUE
C       IVIDX: VEHICLE INDEX
C       IVTYP: VEHICLE TYPE
C       MCVC: MOBILITY CHARACTERISTICS OF VEHICLE
C       NWCAP: NON WHEEL CHAIR CAPACITY
      CALL FIELD(1,TOURLI(ITOUR,1),IVIDX,1,8)
      CALL FIELD(1,TOURLI(ITOUR,1),IVTYP,9,4)
      CALL FIELD(1,VTYPEL(IVTYP),MCVC,1,4)
      CALL FIELD(1,VTYPEL(IVTYP),NWCAP,5,4)
      IF (IVTYP.EQ.IVTYPA) GO TO 290
C       ** STORE DWELL TIME IN KSE **
      CALL FIELD(1,VTYPEL(IVTYP),ROWVCA,17,16)
      DO 275 I1=1,NWCAP
      I2=ROWVCA+I1
      DO 275 J1=1,3
      J2=J1*8+1
      CALL FIELD(1,VCADWE(I2,1),KSE(I1,J1),J2,8)
      J3=J1+3
      CALL FIELD(1,VCADWE(I2,2),KSE(I1,J3),J2,8)
  275 CONTINUE
C       ** CONSTRUCT DWELL TIME **
      DO 285 I1=IVAX1,PA03XX
      CALL FIELD(1,CANDTP(I1,1),I2,6,1)
```

```
      IF (I2.EQ.1) GO TO 285
      PT=0
      DT=0
      DO 280 I3=1,3
      I4=4*I3+1
      CALL FIELD(1,CANDTP(I1,16),J1,I4,4)
      IF (PT.LT.KSE(J1,I3)) PT=KSE(J1,I3)
      I5=I3+3
      IF (DT.LT.KSE(J1,I5)) DT=KSE(J1,I5)
  280 CONTINUE
      CALL FIELD(2,CANDTP(I1,16),PT,17,8)
      CALL FIELD(2,CANDTP(I1,16),DT,25,8)
  285 CONTINUE
  290 CONTINUE
C     ENTER TOUR STARTING TIME
      DO 300 I1=1,PA03VX
C     CHECK TYPE OF BREAK
      CALL FIELD(1,CANDTP(I1,1),I2,1,3)
      IF (I2.NE.2) GO TO 300
      CALL FIELD(1,CANDTP(I1,1),I2,6,1)
      IF (I2.NE.0) GO TO 300
      CALL FIELD(1,CANDTP(I1,2),I2,25,8)
      IF (I2.NE.IVIDX) GO TO 300
C     SET TOUR STARTING POINT AND TIME
      CALL FIELD(2,CANDTP(I1,1),1,6,1)
      IBGNP=I1
      JLAST=I1
      TLAST=CANDTP(I1,12)
      IVWHTR(1,1)=I1
      IVWHTR(1,2)=6
      IVWHTR(1,8)=CANDTP(I1,13)
      CALL FIELD(1,CANDTP(I1,16),IVWHTR(1,7),17,8)
      IVWHTR(1,3)=0
      IVWHTR(1,4)=0
      IVWHTR(1,5)=0
      GO TO 310
  300 CONTINUE
      RETURN
  310 CONTINUE
C     ENTER TOUR BREAK TIME
      DO 330 I1=1,PA03VX
      CALL FIELD(1,CANDTP(I1,1),I2,1,3)
      IF (I2.NE.1) GO TO 330
      CALL FIELD(1,CANDTP(I1,1),I2,6,1)
      IF (I2.NE.0) GO TO 330
      CALL FIELD(1,CANDTP(I1,2),I2,25,8)
      IF (I2.NE.IVIDX) GO TO 330
      I4=NSTPTR+1
      I5=NSTPTR+2
      NSTPTR=I5
      IVWHTR(I4,1)=I1
      IVWHTR(I4,2)=3
      IVWHTR(I5,1)=I1
      IVWHTR(I5,2)=4
      IVWHTR(I4,3)=0
      IVWHTR(I4,4)=0
```

```
      IVWHTR(I4,5)=0
      IVWHTR(I5,3)=0
      IVWHTR(I5,4)=0
      IVWHTR(I5,5)=0
      CALL FIELD(2,CANDTP(I1,1),1,6,1)
      CALL FIELD(1,CANDTP(I1,3),IVWHTR(I4,6),9,12)
      CALL FIELD(1,CANDTP(I1,16),IVWHTR(I4,7),17,8)
      CALL FIELD(1,CANDTP(I1,16),IVWHTR(I5,7),25,8)
      IVWHTR(I4,8)=CANDTP(I1,12)
      IVWHTR(I5,8)=CANDTP(I1,15)
  330 CONTINUE
      NSTPTR=NSTPTR+1
      IVWHTR(NSTPTR,1)=JLAST
      IVWHTR(NSTPTR,2)=5
      IVWHTR(NSTPTR,3)=0
      IVWHTR(NSTPTR,4)=0
      IVWHTR(NSTPTR,5)=0
      IVWHTR(NSTPTR,6)=0
      IVWHTR(NSTPTR,7)=0
      IVWHTR(NSTPTR,8)=TLAST
C     **NEW GAP***NEW SUBTOUR
  350 CONTINUE
      IFLAG3=0
C     SEARCH GAP FOR INSERTION
      I1=IVWHTR(IBGNS,2)
      IF (I1.EQ.1) GO TO 395
      IF (I1.EQ.3) GO TO 360
      IF (I1.EQ.4) GO TO 370
      IF (I1.EQ.5) GO TO 700
      IBGNP=IVWHTR(IBGNS,1)
      GO TO 380
C     SHIFT TO THE LEFT
  360 CONTINUE
      CALL SHIFT(CANDTP,IVWHTR,IBGNS,NSTPTR,PARM03,PARM06)
      IBGNS=IBGNS+1
      IBRK=0
      GO TO 350
  370 CONTINUE
      IBGNP=IVWHTR(IBGNS,1)
      IF (IBRK.EQ.1) IBGNP=IBRKP
      IBRKP=IBGNP
      CALL FIELD(1,CANDTP(IBGNP,5),DRPX,17,16)
      IF (DRPX.NE.0) GO TO 380
      CALL FIELD(1,CANDTP(IBGNP,6),DRPY,17,16)
      IF (DRPY.NE.0) GO TO 380
      IENDS=IBGNS+1
      I2=IBGNS-2
      I3=IVWHTR(I2,1)
      IENDP=IVWHTR(IENDS,1)
      CALL TFETCH(TPD,PA03XX,IENDP,I3,DISTOD)
      IVWHTR(IBGNS,6)=TPD
      CANDTP(IBGNP,5)=CANDTP(I3,5)
      CANDTP(IBGNP,6)=CANDTP(I3,6)
      CALL FIELD(1,CANDTP(I3,5),ITEMP,17,16)
      CALL FIELD(2,CANDTP(IBGNP,5),ITEMP,1,16)
      CALL FIELD(1,CANDTP(I3,6),ITEMP,17,16)
```

```
      CALL FIELD(2,CANDTP(IBGNP,6),ITEMP,1,16)
      IBRK=1
      IBRKP=I3
      GO TO 390
  365 CONTINUE
C     **TRAVEL TIME**
      I1=IBGNS
      I2=IBGNS+1
      I4=IVWHTR(I1,1)
      IF (IVWHTR(I1,2).EQ.4) I4=IBRKP
      I3=IVWHTR(I2,1)
      CALL TFETCH(TPD,PA03XX,I3,I4,DISTOD)
      IVWHTR(I1,6)=TPD
      GO TO 395
  380 IENDS=IBGNS+1
      IENDP=IVWHTR(IENDS,1)
      CALL TFETCH(TPD,PA03XX,IENDP,IBGNP,DISTOD)
      I1=IBGNS
      I2=IBGNS+1
  390 OPENI=IVWHTR(IENDS,8)-IVWHTR(IBGNS,8)-TPD
     1       -IVWHTR(IENDS,7)-IVWHTR(IBGNS,7)
      IF (OPENI.GE.GPSIZE) GO TO 400
  395 IBGNS=IBGNS+1
      IBRK=0
      CALL SHIFT(CANDTP,IVWHTR,IBGNS,NSTPTR,PARM03,PARM06)
C     **IDLE TIME**
      IVWHTR(I1,9)=IVWHTR(I2,8)-IVWHTR(I1,8)-IVWHTR(I1,7)
     1  -IVWHTR(I2,7)-IVWHTR(I1,6)
      GO TO 350
  400 CONTINUE
C     TOUR PRIORITY SETTING
      IPRIOT=0
  410 IPRIOT=IPRIOT+1
      IF (IPRIOT.GT.15) GO TO 365
      IF (PRIOTR(IPRIOT).EQ.0) GO TO 410
      IF (PRIOSM(IPRIOT).EQ.0) GO TO 410
      DO 420 I1=1,15
      PRDIB(I1)=0
  420 CONTINUE
      BIPR=0
      DO 430 J5=IVAX1,PA03XX
      CALL FIELD(1,CANDTP(J5,1),J6,6,1)
      IF (J6.NE.0) GO TO 430
      CANDTP(J5,4)=0
  430 CONTINUE
  440 CONTINUE
      JA=0
      IFLAG3=0
      IFLAG5=0
      PRDMX=0
C     SEARCH FOR THE PATRON WITH THE HIGHEST UPBOUND
      DO 450 I1=IVAX1,PA03XX
      CALL FIELD(1,CANDTP(I1,1),I2,6,1)
      IF (I2.NE.0) GO TO 450
      IF (CANDTP(I1,4).LT.0) GO TO 450
      I2=I1-PA03VX
```

```
      PRDV=PRDBD(I2,IPRIOT)
      IF (PRDMX.GE.PRDV) GO TO 450
      PRDMX=PRDV
      JA=I1
  450 CONTINUE
      IF (PRDMX.LE.BIPR) GO TO 500
C
      NNWCB=0
      NWCCB=0
C     MCPA: MOBILITY CHARACTERISTICS OF PATRON
      CALL FIELD(1,CANDTP(JA,1),MCPA,11,3)
      IF (MCPA.GT.MCVC) GO TO 460
      CALL FIELD(1,CANDTP(JA,2),IPRIOP,5,4)
C     CHECK FOR COMPATIBLE TOUR PRIORITY
      IF (PRIOTR(IPRIOP).NE.1) GO TO 460
C     CHECK TIME WINDOWS
      CALL FIELD(1,CANDTP(JA,3),ICSBTR(1,6),9,12)
      CALL FIELD(1,CANDTP(JA,16),ICSBTR(1,7),17,8)
      IF (IVWHTR(IBGNS,2).EQ.4) IBGNP=IBRKP
      CALL TFETCH(TPD,PA03XX,JA,IBGNP,DISTOD)
      TALSB=IVWHTR(IBGNS,8)+TPD+ICSBTR(1,7)+IVWHTR(IBGNS,7)
      IF (TALSB.GT.CANDTP(JA,12)) GO TO 460
      ICSBTR(1,8)=CANDTP(JA,12)
      ICSBTR(2,8)=ICSBTR(1,8)+ICSBTR(1,6)
      CALL TFETCH(ICSBTR(2,6),PA03XX,IENDP,JA,DISTOD)
      CALL FIELD(1,CANDTP(JA,16),ICSBTR(2,7),25,8)
      T1=ICSBTR(2,8)+ICSBTR(2,6)+ICSBTR(2,7)+IVWHTR(IENDS,7)
      IF (T1.LE.IVWHTR(IENDS,8)) GO TO 455
      T2=T1-IVWHTR(IENDS,8)
      ICSBTR(2,8)=ICSBTR(2,8)-T2
      ICSBTR(1,8)=ICSBTR(1,8)-T2
      IF (ICSBTR(1,8).LT.CANDTP(JA,10)) GO TO 460
      IF (ICSBTR(2,8).LT.CANDTP(JA,13)) GO TO 460
  455 CONTINUE
C     CHECK VEHICLE CAPACITY
      CALL FIELD(1,VTYPEL(IVTYP),JRDWT,17,16)
      CALL FIELD(1,CANDTP(JA,16),NNWP,1,8)
      CALL FIELD(1,CANDTP(JA,16),NLFP,9,4)
      NA=NNWCB+NNWP+NLFP
      IF (NA.GT.NWCAP) GO TO 460
      CALL FIELD(1,CANDTP(JA,16),NWCP,13,4)
      NB=NWCCB+NWCP
      N1=JRDWT+NA
      CALL FIELD(1,VCADWE(N1,1),NWCCP,1,8)
      IF (NB.LE.NWCCP) GO TO 470
  460 CANDTP(JA,4)=-PRDBD(JA-PA03VX,IPRIOT)
      GO TO 440
  470 CONTINUE
      I4=PA03PX/31
      I5=PA03PX-31*I4
      IF (I5.NE.0) I4=I4+1
      I4=I4+4
      I6=JA-PA03VX
      DO 490 I1=1,I4
      I2=I4*(I6-1)+I1
      BVECT1(I1)=BASETP(I2)
```

```
  490 CONTINUE
      CALL SBTOUR(JA,TALSB,IPRIOP,IFLAG3,IFLAG5,BIPR,
     1 BVECT1,CANDTP,COMPAT,DISTOD,DISTOO,IBSBTR,ICSBTR,ISE,
     2 IVWHTR,KSE,PRDCOD,PRIOTR,VCADWE,PRDIB,PRDIC,
     3 IBGNP,IBGNS,IBSTP,PA03XX,IENDP,IENDS,PA03PX,IPRIOT,PA03VX,JRDWT,
     4 MCVC,PA08UX,NWCAP,
     5 PARM03,PARM04,PARM05,PARM06,PARM10)
C      IFLAG3=1: A BETTER SUBTOUR IS FOUND
C      IFLAG3=0: OTHERWISE
C      SEE FESIBL FOR IFLAG5
      IF (IFLAG5.EQ.1) GO TO 495
      PRDBD(JA-PA03VX,IPRIOT)=BIPR
  495 CONTINUE
      IF (IFLAG3.EQ.1) IFLAG4=1
C      IFLAG4=1 IF IFLAG3=1
      GO TO 440
  500 IF (IFLAG4.EQ.0) GO TO 410
      IFLAG3=0
      IFLAG4=0
  600 CONTINUE
C     **UPDATE BASETP AND COMPAT (SUM AND RIDE SHARES)
      DO 620 I1=1,IBSTP
      IF (IBSBTR(I1,2).NE.1) GO TO 620
      I2=IBSBTR(I1,1)
      CALL FIELD(2,CANDTP(I2,1),1,6,1)
      CANDTP(I2,4)=1
      I2=I2-PA03VX
      I3=I2/31
      I4=I2-31*I3
      IF (I4.EQ.0) I4=31
      IF (I4.NE.31) I3=I3+1
      I3=I3+4
      I4=I4+1
      J3=PA03PX/31
      J4=PA03PX-31*J3
      IF (J4.NE.0) J3=J3+1
      J3=J3+4
      IPR=IBSBTR(I1,12)
      PRIOSM(IPR)=PRIOSM(IPR)-1
      J6=IPR/4
      J7=IPR-4*J6
      IF (J7.EQ.0) J7=4
      IF (J7.NE.4) J6=J6+1
      J7=(J7-1)*8+1
      DO 620 K1=IVAX1,PA03XX
      K2=K1-PA03VX
      K3=(K2-1)*J3+I3
      CALL FIELD(1,BASETP(K3),L1,I4,1)
      IF (L1.EQ.0) GO TO 620
      CALL FIELD(2,BASETP(K3),0,I4,1)
      K3=(K2-1)*J3+J6
      CALL FIELD(1,BASETP(K3),L1,J7,8)
      L1=L1-1
      CALL FIELD(2,BASETP(K3),L1,J7,8)
      CALL FIELD(1,COMPAT(K3),L1,J7,8)
      L1=L1-IBSBTR(I1,9)-IBSBTR(I1,10)-IBSBTR(I1,11)
```

```
      CALL FIELD(2,COMPAT(K3),L1,J7,8)
  620 CONTINUE
      NSTPTR=NSTPTR+IBSTP
C     SHIFTING POINTERS FOR PREVIOUSLY ASSIGNED STOPS
      I1=IBGNS+IBSTP+1
      DO 640 I2=NSTPTR,I1,-1
      I3=I2-IBSTP
      DO 640 I4=1,8
      IVWHTR(I2,I4)=IVWHTR(I3,I4)
  640 CONTINUE
C     FILLING IN POINTERS FOR NEW STOPS
      I1=IBGNS+1
      I2=IBGNS+IBSTP
      DO 660 I3=I1,I2
      I4=I3-IBGNS
      DO 660 I5=1,8
      IVWHTR(I3,I5)=IBSBTR(I4,I5)
  660 CONTINUE
C     ***IDLE TIME***
      I1=IBGNS
      I2=IBGNS+1
      I4=IVWHTR(I1,1)
      I3=IVWHTR(I2,1)
      IF (IVWHTR(I2,2).EQ.5) GO TO 670
      IF (IVWHTR(I1,2).EQ.4) I4=IBRKP
      CALL TFETCH(TPD,PA03XX,I3,I4,DISTOD)
      IVWHTR(I1,6)=TPD
      IVWHTR(I1,9)=IVWHTR(I2,8)-IVWHTR(I1,8)-IVWHTR(I1,7)
     1  -IVWHTR(I2,7)-IVWHTR(I1,6)
  670 CONTINUE
      IF (ISUBTR.EQ.0) GO TO 350
      I1=NSTPTR-1
      I2=NSTPTR
      GO TO 710
  700 CONTINUE
C     ***SHIFT ENDING TIME TO THE LEFT***
      I1=IBGNS-1
      I2=IBGNS
  710 CONTINUE
      I3=IVWHTR(I2,1)
      IVWHTR(I2,10)=CANDTP(I3,12)-IVWHTR(I2,8)
      I4=IVWHTR(I1,1)
      IF (IVWHTR(I1,2).EQ.4) I4=IBRKP
      CALL TFETCH(TPD,PA03XX,I3,I4,DISTOD)
      IVWHTR(I1,6)=TPD
      IVWHTR(I2,8)=IVWHTR(I1,8)+IVWHTR(I1,6)+IVWHTR(I1,7)
     1 +IVWHTR(I2,7)
      IF (CANDTP(I3,10).GT.IVWHTR(I2,8)) IVWHTR(I2,8)=CANDTP(I3,10)
C     ***IDLE TIME***
      IVWHTR(I1,9)=IVWHTR(I2,8)-IVWHTR(I1,8)-IVWHTR(I1,7)
     1  -IVWHTR(I1,6)-IVWHTR(I2,7)
C     **SHIFT STARTING TIME TO THE RIGHT**
      I1=IVWHTR(1,1)
      IVWHTR(1,10)=CANDTP(I1,15)-IVWHTR(1,8)
      T1=IVWHTR(2,8)-IVWHTR(1,6)-IVWHTR(2,7)
      IF (T1.LE.CANDTP(I1,12)) IVWHTR(1,8)=T1
```

```
      IF (T1.GT.CANDTP(I1,15)) IVWHTR(1,8)=CANDTP(I1,15)
      IVWHTR(1,9)=IVWHTR(2,8)-IVWHTR(1,8)
     1  -IVWHTR(1,6)-IVWHTR(2,7)
      DO 740 I1=1,NSTPTR
      I2=IVWHTR(I1,1)
C     TRIP STATUS
      CALL FIELD(2,CANDTP(I2,1),1,6,1)
C     TOUR
      CALL FIELD(2,CANDTP(I2,3),PA08UX,1,8)
      I3=IVWHTR(I1,2)
      IF (I3.EQ.2) GO TO 730
      IF (I3.EQ.4) GO TO 730
      IF (I3.EQ.6) GO TO 730
      CALL FIELD(2,CANDTP(I2,1),I1,17,8)
      CALL FIELD(2,CANDTP(I2,4),IVWHTR(I1,8),1,16)
      CANDTP(I2,11)=IVWHTR(I1,8)
      IVWHTR(I1,10)=CANDTP(I2,12)-CANDTP(I2,11)
      GO TO 740
  730 CANDTP(I2,14)=IVWHTR(I1,8)
      IVWHTR(I1,10)=CANDTP(I2,15)-CANDTP(I2,14)
      CALL FIELD(2,CANDTP(I2,1),I1,25,8)
      CALL FIELD(2,CANDTP(I2,4),IVWHTR(I1,8),17,16)
  740 CONTINUE
      DO 750 I1=1,NSTPTR
      I2=PA09XX+I1
      CALL FIELD(2,TSTOPL(I2,1),IVWHTR(I1,2),1,4)
      CALL FIELD(2,TSTOPL(I2,1),IVWHTR(I1,7),5,8)
      CALL FIELD(2,TSTOPL(I2,1),IVWHTR(I1,1),13,16)
      CALL FIELD(2,TSTOPL(I2,2),IVWHTR(I1,3),1,8)
      CALL FIELD(2,TSTOPL(I2,2),IVWHTR(I1,4),9,8)
      CALL FIELD(2,TSTOPL(I2,2),IVWHTR(I1,5),17,8)
      I3=IVWHTR(I1,6)/8
      CALL FIELD(2,TSTOPL(I2,3),I3,1,8)
      I3=IVWHTR(I1,9)/8
      CALL FIELD(2,TSTOPL(I2,3),I3,9,12)
      I3=IVWHTR(I1,10)/8
      CALL FIELD(2,TSTOPL(I2,3),I3,21,12)
  750 CONTINUE
      CALL FIELD(2,TOURLI(PA08UX,2),NSTPTR,5,12)
      I1=PA09XX+1
      CALL FIELD(2,TOURLI(PA08UX,2),I1,17,16)
      WRITE(6,770) PA08UX,NSTPTR
  770 FORMAT( ' TOUR #',I3,' IS COMPLETED : ',I4,' STOPS')
      PA09XX=PA09XX+NSTPTR
      GO TO 200
      END
```

```
      SUBROUTINE PRDCTY(IPCODE,PATRN,TRIP,PATRT,TRPRT,TIME,PRDV)
      IMPLICIT INTEGER (B-Y)
      IF (IPCODE.EQ.1) GO TO 100
      IF (IPCODE.EQ.2) GO TO 200
      IF (IPCODE.EQ.3) GO TO 300
      IF (IPCODE.EQ.4) GO TO 400
      IF (IPCODE.EQ.5) GO TO 500
      IF (IPCODE.EQ.6) GO TO 600
  100 PRDV=PATRN
      RETURN
  200 PRDV=TRIP
      RETURN
  300 PRDV=(PATRN*3840)/TIME
      RETURN
  400 PRDV=(TRIP*3840)/TIME
      RETURN
  500 PRDV=(PATRT*480)/TIME
      RETURN
  600 PRDV=(TRPRT*480)/TIME
      RETURN
      END
```

```
      SUBROUTINE PREPRD(IP,CANDTP,ICSBTR,PRDCOD,PRDIC,ICSTP,PARM03,
     1 PARM05)
      IMPLICIT INTEGER (B-Y)
      DIMENSION CANDTP(PARM03,1),ICSBTR(PARM05,1),PRDCOD(1),PRDIC(1)
C     PRDCOD(IP):
C     1:NUMBER OF PATRONS WITH PRIORITY IP
C     2:NUMBER OF TRIPS WITH PRIORITY IP
C     3:PATRONS PER UNIT TIME
C     4:TRIPS PER UNIT TIME
C     5: (PATRN*ERT)/TIME
C     6: (TRIP*ERT)/TIME
      PATRN=0
      TRIP=0
      PATRT=0
      TRPRT=0
      I1=PRDCOD(IP)
      IF (I1.EQ.1) GO TO 100
      IF (I1.EQ.2) GO TO 200
      IF (I1.EQ.3) GO TO 300
      IF (I1.EQ.4) GO TO 400
      IF (I1.EQ.5) GO TO 500
      IF (I1.EQ.6) GO TO 600
  100 CONTINUE
      DO 150 I2=1,ICSTP
      IF (ICSBTR(I2,12).NE.IP) GO TO 150
      IF (ICSBTR(I2,2).NE.1) GO TO 150
      PATRN=PATRN+ICSBTR(I2,9)+ICSBTR(I2,10)+ICSBTR(I2,11)
  150 CONTINUE
      CALL PRDCTY(I1,PATRN,TRIP,PATRT,TRPRT,TIME,PRDIC(IP))
      RETURN
  200 CONTINUE
      DO 250 I2=1,ICSTP
      IF (ICSBTR(I2,12).NE.IP) GO TO 250
      IF (ICSBTR(I2,2).NE.1) GO TO 250
      TRIP=TRIP+1
  250 CONTINUE
      CALL PRDCTY(I1,PATRN,TRIP,PATRT,TRPRT,TIME,PRDIC(IP))
      RETURN
  300 CONTINUE
      DO 350 I2=1,ICSTP
      IF (ICSBTR(I2,12).NE.IP) GO TO 350
      IF (ICSBTR(I2,2).NE.1) GO TO 350
      PATRN=PATRN+ICSBTR(I2,9)+ICSBTR(I2,10)+ICSBTR(I2,11)
  350 CONTINUE
      TIME=ICSBTR(ICSTP,8)-ICSBTR(1,8)
      CALL PRDCTY(I1,PATRN,TRIP,PATRT,TRPRT,TIME,PRDIC(IP))
      RETURN
  400 CONTINUE
      DO 450 I2=1,ICSTP
      IF (ICSBTR(I2,12).NE.IP) GO TO 450
      IF (ICSBTR(I2,2).NE.1) GO TO 450
      TRIP=TRIP+1
  450 CONTINUE
      TIME=ICSBTR(ICSTP,8)-ICSBTR(1,8)
      CALL PRDCTY(I1,PATRN,TRIP,PATRT,TRPRT,TIME,PRDIC(IP))
      RETURN
```

```
  500 CONTINUE
      DO 550 I2=1,ICSTP
      IF (ICSBTR(I2,12).NE.IP) GO TO 550
      IF (ICSBTR(I2,2).NE.1) GO TO 550
      I3=ICSBTR(I2,9)+ICSBTR(I2,10)+ICSBTR(I2,11)
      J1=ICSBTR(I2,1)
      CALL FIELD(1,CANDTP(J1,3),I4,9,12)
      PATRT=PATRT+I3*I4
  550 CONTINUE
      TIME=ICSBTR(ICSTP,8)-ICSBTR(1,8)
      CALL PRDCTY(I1,PATRN,TRIP,PATRT,TRPRT,TIME,PRDIC(IP))
      RETURN
  600 DO 650 I2=1,ICSTP
      IF (ICSBTR(I2,12).NE.IP) GO TO 650
      IF (ICSBTR(I2,2).NE.1) GO TO 650
      J1=ICSBTR(I2,1)
      CALL FIELD(1,CANDTP(J1,3),I3,9,12)
      TRPRT=TRPRT+I3
  650 CONTINUE
      TIME=ICSBTR(ICSTP,8)-ICSBTR(1,8)
      CALL PRDCTY(I1,PATRN,TRIP,PATRT,TRPRT,TIME,PRDIC(IP))
      RETURN
      END
```

```
      SUBROUTINE READCV(PA03XX,PA03PX,PA08XX,PA08RX,PA03VX,BASETP,
     1 CANDTP,COMPAT,DISTOD,DISTOO,PRIORL,TOURLI,
     2 VCADWE,VTYPEL, PARM03,PARM08,PARM10)
      IMPLICIT INTEGER (B-Y)
      DIMENSION BASETP(1),CANDTP(PARM03,1)
      DIMENSION COMPAT(1),DISTOD(1),DISTOO(1)
      DIMENSION PRIORL(1),TOURLI(PARM08,1)
      DIMENSION VCADWE(PARM10,1),VTYPEL(1)
      READ(11,1001) I,I,PA03XX,I,I,I,PA07XX,PA08XX,I,PA10XX,
     1  I,I,I,I,I,I,I,I,PA03VX,PA08RX,IBX,IDX
 1001 FORMAT(20I6)
      PA03PX=PA03XX-PA03VX
      READ(11,1071) (VTYPEL(I),I=1,PA07XX)
 1071 FORMAT(80I2)
      DO 1080 I=1,PA10XX
      READ(11,1081) M1, (VCADWE(I,J),J=1,2)
 1080 CONTINUE
 1081 FORMAT(I5,80I2/5X,80I2)
      DO 1110 I=1,PA08XX
      READ(11,1081) M1, (TOURLI(I,J),J=1,3)
 1110 CONTINUE
      DO 1140 I=1,PA03XX
      READ(11,1081) M1, (CANDTP(I,J),J=1,16)
 1140 CONTINUE
      I1=1
      I2=IBX
      DO 1170 I=1,PA03PX
      READ(11,1081) M1, (BASETP(J),J=I1,I2)
      READ(11,1081) M1, (COMPAT(J),J=I1,I2)
      I1=I1+IBX
      I2=I2+IBX
 1170 CONTINUE
      DO 1200 I=1,IDX
      READ(11,1081) M1, DISTOO(I),DISTOD(I)
 1200 CONTINUE
      DO 1220 I=1,15
      READ(11,1081) M1, PRIORL(I)
 1220 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RMTIME(IA,IK2,JK,IFLAG2,
     1 CANDTP,ISE,KSE,IC1,IC2,RTB,RTS,SHTB,SHTS,
     2 PARM03,PARM05 )
      IMPLICIT INTEGER (B-Y)
      DIMENSION CANDTP(PARM03,1),ISE(PARM05,1),KSE(PARM05,1)
C
C     IFLAG2=1: A BETTER SUBTOUR OF IK2 STOPS IS FOUND
C     IFLAG2=0: OTHERWISE
      DO 100 I1=1,IK2
      IF (KSE(I1,1).EQ.IC1.AND.KSE(I1,2).EQ.1) I2=I1
      IF (KSE(I1,1).EQ.IC1.AND.KSE(I1,2).EQ.2) I3=I1
  100 CONTINUE
      CALL FIELD(1,CANDTP(IC1,3),TMXOB,21,12)
      RTS=TMXOB-(KSE(I3,8)-KSE(I2,8))
      SHTS=KSE(IK2,8)-KSE(1,8)
      IF (RTB-RTS) 300,200,500
  200 IF (SHTB.LE.SHTS) GO TO 500
  300 RTB=RTS
      SHTB=SHTS
      IFLAG2=1
      JK=IK2
      IC2=IA
      DO 400 I1=1,IK2
      DO 400 I2=1,12
      ISE(I1,I2)=KSE(I1,I2)
  400 CONTINUE
  500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SBTGEN(JA,IP1,IFLAG2,IFLAG5,
     1 BVECT1,CANDTP,COMPAT,DISTCD,DISTOO,ICSBTR,ISE,IVWHTR,
     2 KSE,VCADWE,
     3 IBGNP,IBGNS,IC1,PA03XX,ICSTP,IENDP,IENDS,PA03PX,PA03VX,
     4 JRDWT,MCVC,PA08UX,NWCAP,
     5 PARM03,PARM04,PARM05,PARM06,PARM10)
      IMPLICIT INTEGER (B-Y)
      DIMENSION BVECT1(1),CANDTP(PARM03,1),COMPAT(1)
      DIMENSION DISTCD(1),DISTOO(1),ICSBTR(PARM05,1)
      DIMENSION ISE(PARM05,1),IVWHTR(PARM06,1),KSE(PARM05,1)
      DIMENSION VCADWE(PARM10,1)
      DIMENSION BVECT2(20)
      JK=0
      IVAX1=PA03VX+1
  200 IFLAG2=0
      IC2=0
      RTS=0
      RTB=0
      SHTS=99999
      SHTB=99999
C     SEARCH A PATRON FOR RIDE SHARING
      DO 290 JB=IVAX1,PA03XX
      IF (JB.EQ.JA) GO TO 290
      CALL FIELD(1,CANDTP(JB,1),I1,6,1)
      IF (I1.NE.0) GO TO 290
      CALL FIELD(1,CANDTP(JB,1),MCPA,11,3)
      IF (MCPA.GT.MCVC) GO TO 290
      CALL FIELD(1,CANDTP(JB,2),IPRIOP,5,4)
      IF (IPRIOP.NE.IP1) GO TO 290
C     CAN THE PATRON SHARE RIDE WITH THE BASE TRIP?
      I1=JB-PA03VX
      J1=I1/31
      J2=I1-31*J1
      IF (J2.EQ.0) J2=31
      IF (J2.NE.31) J1=J1+1
      J1=J1+4
      J2=J2+1
      CALL FIELD(1,BVECT1(J1),I2,J2,1)
      IF (I2.EQ.0) GO TO 290
C     **TRANSFER COMPATIBILITY MATRIX**
      I4=PA03PX/31
      I5=PA03PX-31*I4
      IF (I5.NE.0) I4=I4+1
      I5=I4+4
      I3=JB-PA03VX
      DO 240 I1=1,I4
      I2=I5*(I3-1)+I1+4
      BVECT2(I1)=COMPAT(I2)
  240 CONTINUE
C     **TEST FOR RIDE COMPABILITY**
      DO 280 I1=1,ICSTP
      IF (ICSBTR(I1,1).EQ.JA) GO TO 280
      IF (ICSBTR(I1,1).EQ.JB) GO TO 280
      IF (ICSBTR(I1,2).NE.1) GO TO 280
      I2=ICSBTR(I1,1)-PA03VX
      J1=I2/31
```

```
      J2=I2-31*J1
      IF (J2.EQ.0) J2=31
      IF (J2.NE.31) J1=J1+1
      J2=J2+1
      CALL FIELD(1,BVECT2(J1),I3,J2,1)
      IF (I3.EQ.0) GO TO 290
  280 CONTINUE
      CALL TRCOMB(JB,JK,IFLAG2,IFLAG5,
     1 CANDTP,DISTOD,DISTOO,ISE,ICSBTR,IVWHTR,KSE,VCADWE,
     2 IBGNP,IBGNS,IC1,IC2,PA03XX,ICSTP,IENDP,IENDS,JRDWT,
     3 NWCAP,RTB,RTS,SHTB,SHTS,
     4 PARM03,PARM04,PARM05,PARM06,PARM10)
  290 CONTINUE
      IF (IFLAG2.EQ.0) RETURN
      ICSTP=JK
      DO 400 I1=1,JK
      DO 400 I2=1,12
      ICSBTR(I1,I2)=ISE(I1,I2)
  400 CONTINUE
C     REMOVE 1 FROM BVECT1
      I1=IC2-PA03VX
      J1=I1/31
      J2=I1-31*J1
      IF (J2.EQ.0) J2=31
      IF (J2.NE.31) J1=J1+1
      J1=J1+4
      J2=J2+1
      CALL FIELD(2,BVECT1(J1),0,J2,1)
C     REMOVE 1 FROM PRIORITY SUM FROM BVECT1
      J1=IP1/4
      J2=IP1-4*J1
      IF (J2.EQ.0) J2=4
      IF (J2.NE.4) J1=J1+1
      IF (J2.EQ.4) BVECT1(J1)=BVECT1(J1)-1
      IF (J2.EQ.3) BVECT1(J1)=BVECT1(J1)-256
      IF (J2.EQ.2) BVECT1(J1)=BVECT1(J1)-65536
      IF (J2.EQ.1) BVECT1(J1)=BVECT1(J1)-16777216
      GO TO 200
      END
```

```
      SUBROUTINE SBTOUR(JA,TALSB,IPRIOP,IFLAG3,IFLAG5,BIPR,
     1 BVECT1,CANDTP,COMPAT,DISTOD,DISTOO,IBSBTR,ICSBTR,ISE,
     2 IVWHTR,KSE,PRDCOD,PRIOTR,VCADWE,PRDIB,PRDIC,
     3 IBGNP,IBGNS,IBSTP,PA03XX,IENDP,IENDS,PA03PX,IPRIOT,PA03VX,JRDWT,
     4 MCVC,PA08UX,NWCAP,
     5 PARM03,PARM04,PARM05,PARM06,PARM10)
      IMPLICIT INTEGER (B-Y)
      DIMENSION BVECT1(1),CANDTP(PARM03,1),COMPAT(1),DISTOD(1)
      DIMENSION DISTOO(1),IBSBTR(PARM05,1),ICSBTR(PARM05,1)
      DIMENSION ISE(PARM05,1),IVWHTR(PARM06,1),KSE(PARM05,1)
      DIMENSION PRDCOD(1),PRIOTR(1),VCADWE(PARM10,1)
      DIMENSION PRDIB(1),PRDIC(1)
      ICSTP=2
      IC1=JA
      IP1=IPRIOT
      DO 80 I1=1,15
      PRDIC(I1)=0
   80 CONTINUE
      ICSBTR(1,1)=JA
      ICSBTR(1,2)=1
      CALL FIELD(1,CANDTP(JA,16),ICSBTR(1,9),1,8)
      CALL FIELD(1,CANDTP(JA,16),ICSBTR(1,10),9,4)
      CALL FIELD(1,CANDTP(JA,16),ICSBTR(1,11),13,4)
      ICSBTR(1,3)=ICSBTR(1,9)
      ICSBTR(1,4)=ICSBTR(1,10)
      ICSBTR(1,5)=ICSBTR(1,11)
      ICSBTR(1,12)=IPRIOP
      ICSBTR(2,1)=JA
      ICSBTR(2,2)=2
      CALL FIELD(1,CANDTP(JA,16),ICSBTR(2,7),25,8)
      ICSBTR(2,3)=0
      ICSBTR(2,4)=0
      ICSBTR(2,5)=0
      ICSBTR(2,9)=ICSBTR(1,9)
      ICSBTR(2,10)=ICSBTR(1,10)
      ICSBTR(2,11)=ICSBTR(1,11)
      ICSBTR(2,12)=ICSBTR(1,12)
      X1=CANDTP(JA,13)-ICSBTR(1,6)
      IF (TALSB.GT.X1) X1=TALSB
      CALL FIELD(1,CANDTP(JA,3),EMXRT,21,12)
      X2=CANDTP(JA,15)-EMXRT
      IF (TALSB.GT.X2) EMXRT=CANDTP(JA,15)-TALSB
  100 CONTINUE
C     CALL SUBTOUR GENERATOR
      CALL SBTGEN(JA,IP1,IFLAG2,IFLAG5,
     1 BVECT1,CANDTP,COMPAT,DISTOD,DISTOO,ICSBTR,ISE,IVWHTR,
     2 KSE,VCADWE,
     3 IBGNP,IBGNS,IC1,PA03XX,ICSTP,IENDP,IENDS,PA03PX,PA03VX,
     4 JRDWT,MCVC,PA08UX,NWCAP,
     5 PARM03,PARM04,PARM05,PARM06,PARM10)
C     CALCULATE PRODUCTIVITY OF ICSBTR
      CALL PREPRD(IP1,CANDTP,ICSBTR,PRDCOD,PRDIC,ICSTP,PARM03,
     1 PARM05)
      IF (PRDIC(IP1)-PRDIB(IP1)) 600,140,200
  140 IF (IP1.NE.IPRIOT) GO TO 160
  150 IF (PRDIB(IP1).EQ.0) RETURN
```

```
  160 IP1=IP1+1
      IF (IP1.GT.15) GO TO 600
      IF (PRIOTR(IP1).NE.1) GO TO 160
      GO TO 100
  200 CONTINUE
      DO 220 I1=IPRIOT,IP1
      PRDIB(I1)=PRDIC(I1)
  220 CONTINUE
      BIPR=PRDIC(IPRIOT)
      IP1=IP1+1
      IF (IP1.GT.15) GO TO 320
      DO 300 IP2=IP1,15
      IF (PRIOTR(IP2).NE.1) GO TO 300
      J1=IP2/4
      J2=IP2-4*J1
      IF (J2.EQ.0) J2=4
      IF (J2.NE.4) J1=J1+1
      J2=(J2-1)*8+1
      CALL FIELD(1,BVECT1(J1),I1,J2,8)
      IF (I1.EQ.0) GO TO 300
      CALL SBTGEN(JA,IP2,IFLAG2,IFLAG5,
     1 BVECT1,CANDTP,COMPAT,DISTOD,DISTOO,ICSBTR,ISE,IVWHTR,
     2 KSE,VCADWE,
     3 IBGNP,IBGNS,IC1,PA03XX,ICSTP,IENDP,IENDS,PA03PX,PA03VX,
     4 JRDWT,MCVC,PA08UX,NWCAP,
     5 PARM03,PARM04,PARM05,PARM06,PARM10)
      CALL PREPRD(IP2,CANDTP,ICSBTR,PRDCOD,PRDIC,ICSTP,PARM03,
     1 PARM05)
      PRDIB(IP2)=PRDIC(IP2)
  300 CONTINUE
C     A SUBTOUR IS MADE
  320 IBSTP=ICSTP
      IFLAG3=1
      DO 400 I1=1,ICSTP
      DO 400 I2=1,12
      IBSBTR(I1,I2)=ICSBTR(I1,I2)
  400 CONTINUE
      IENDP=IVWHTR(IENDS,1)
      I1=ICSBTR(ICSTP,1)
      CALL TFETCH(TPD,PA03XX,IENDP,I1,DISTOD)
      ICSBTR(ICSTP,6)=TPD
  600 RETURN
      END
```

```
      SUBROUTINE SHIFT(CANDTP,IVWHTR,IBGNS,NSTPTR,PARM03,PARM06)
      IMPLICIT INTEGER (B-Y)
      DIMENSION CANDTP(PARM03,1),IVWHTR(PARM06,1)
C     SHIFT TO THE LEFT
C     IS THE SHIFTING POSSIBLE?
      I1=IBGNS-1
      I2=IBGNS
      L1=IVWHTR(I1,8)+IVWHTR(I1,6)+IVWHTR(I1,7)+IVWHTR(I2,7)
      DELTM=IVWHTR(I2,8)-L1
C     DETERMINE SUBTOUR'S LENGTH
      DO 350 I3=I2,NSTPTR
      IF (IVWHTR(I3,2).EQ.1) GO TO 350
      IF (IVWHTR(I3,2).EQ.3) GO TO 360
      IF (IVWHTR(I3,2).EQ.4) GO TO 400
      IF (IVWHTR(I3,2).EQ.5) GO TO 400
      IF (IVWHTR(I3,3).NE.0) GO TO 350
      IF (IVWHTR(I3,4).NE.0) GO TO 350
      IF (IVWHTR(I3,5).NE.0) GO TO 350
      GO TO 400
  350 CONTINUE
      RETURN
  360 I3=I3+1
  400 CONTINUE
      I4=I3-I2
      IF (DELTM.LE.0) GO TO 600
C     SHIFT TO THE LEFT
      DO 450 J1=I2,I3
      J2=IVWHTR(J1,1)
      J3=IVWHTR(J1,2)
      IF (J3.EQ.2) GO TO 420
      IF (J3.EQ.4) GO TO 420
      IF (J3.EQ.6) GO TO 420
      DELTA=IVWHTR(J1,8)-CANDTP(J2,10)
      GO TO 430
  420 DELTA=IVWHTR(J1,8)-CANDTP(J2,13)
  430 CONTINUE
      IF (DELTA.GE.DELTM) GO TO 450
      DELTM=DELTA
  450 CONTINUE
      DO 500 J1=I2,I3
      IVWHTR(J1,8)=IVWHTR(J1,8)-DELTM
  500 CONTINUE
  600 IBGNS=IBGNS+I4
      RETURN
      END
```

```
      SUBROUTINE TFETCH (I,J,K,L,IV)
      DIMENSION IV(1)
      L1=(K-1)*J+L
      L2=L1/3
      L3=L1-3*L2
      IF (L3.NE.0)  L2=L2+1
      IF (L3.EQ.0)  L3=3
      L3=10*L3-7
      CALL FIELD (1,IV(L2),I,L3,10)
      RETURN
      END
```

```
      SUBROUTINE TRCOMB(IA,JK,IFLAG2,IFLAG5,
     1 CANDTP,DISTOD,DISTOO,ISE,ICSBTR,IVWHTR,KSE,VCADWE,
     2 IBGNP,IBGNS,IC1,IC2,PA03XX,ICSTP,IENDP,IENDS,JRDWT,
     3 NWCAP,RTB,RTS,SHTB,SHTS,
     4 PARM03,PARM04,PARM05,PARM06,PARM10)
      IMPLICIT INTEGER (B-Y)
      DIMENSION CANDTP(PARM03,1),DISTOD(1),DISTOO(1)
      DIMENSION ISE(PARM05,1),ICSBTR(PARM05,1),IVWHTR(PARM06,1)
      DIMENSION KSE(PARM05,1),VCADWE(PARM10,1)
      IK=ICSTP
      IK1=IK+1
      IK2=IK+2
      DO 100 I1=1,IK
      I2=ICSBTR(I1,1)
      I3=ICSBTR(I1,2)
      IF (I2.EQ.IC1.AND.I3.EQ.2) IJ=I1
  100 CONTINUE
      II=2
      IL=3
      IM=IK2
C     PICK LIMIT
      DO 500 J7=II,IJ
C     DROP LIMIT
      DO 400 J8=IL,IM
      IF (J7.GE.J8) GO TO 400
      KSE(IK1,1)=IA
      KSE(IK1,2)=1
      KSE(IK2,1)=IA
      KSE(IK2,2)=2
      CALL FIELD(1,CANDTP(IA,2),KPRIO,5,4)
      CALL FIELD(1,CANDTP(IA,16),NNWP,1,8)
      CALL FIELD(1,CANDTP(IA,16),NLFP,9,4)
      CALL FIELD(1,CANDTP(IA,16),NWCP,13,4)
      KSE(IK2,9)=NNWP
      KSE(IK2,10)=NLFP
      KSE(IK2,11)=NWCP
      KSE(IK2,12)=KPRIO
      KSE(IK1,12)=KPRIO
      J3=1
      DO 130 J4=1,IK
      IF (J7.NE.J4) GO TO 110
      IF (J7.EQ.IK1.AND.J8.EQ.IK2) GO TO 120
      KSE(J3,1)=IA
      KSE(J3,2)=1
      KSE(J3,9)=NNWP
      KSE(J3,10)=NLFP
      KSE(J3,11)=NWCP
      KSE(J3,12)=KPRIO
      J3=J3+1
  110 IF (J8.NE.J4+1) GO TO 120
      KSE(J3,1)=IA
      KSE(J3,2)=2
      KSE(J3,9)=NNWP
      KSE(J3,10)=NLFP
      KSE(J3,11)=NWCP
      KSE(J3,12)=KPRIO
```

```
      J3=J3+1
  120 KSE(J3,1)=ICSBTR(J4,1)
      KSE(J3,2)=ICSBTR(J4,2)
      KSE(J3,9)=ICSBTR(J4,9)
      KSE(J3,10)=ICSBTR(J4,10)
      KSE(J3,11)=ICSBTR(J4,11)
      KSE(J3,12)=ICSBTR(J4,12)
      J3=J3+1
  130 CONTINUE
C     CALCULATE NUMBER OF PATRON ON BOARD
      J2=0
      DO 180 J1=1,IK2
      IF (J1.NE.1) GO TO 150
      KSE(J1,3)=KSE(J1,9)
      KSE(J1,4)=KSE(J1,10)
      KSE(J1,5)=KSE(J1,11)
      GO TO 180
  150 CONTINUE
      J2=J2+1
      J3=KSE(J1,2)
      IF (J3.EQ.2) GO TO 170
      KSE(J1,3)=KSE(J2,3)+KSE(J1,9)
      KSE(J1,4)=KSE(J2,4)+KSE(J1,10)
      K1=KSE(J1,3)+KSE(J1,4)
      IF (K1.GT.NWCAP) GO TO 400
      KSE(J1,5)=KSE(J2,5)+KSE(J1,11)
      N1=JRDWT+K1
      CALL FIELD(1,VCADWE(N1,1),NWCCP,1,8)
      IF (KSE(J1,5).GT.NWCCP) GO TO 400
      GO TO 180
  170 KSE(J1,3)=KSE(J2,3)-KSE(J1,9)
      KSE(J1,4)=KSE(J2,4)-KSE(J1,10)
      KSE(J1,5)=KSE(J2,5)-KSE(J1,11)
  180 CONTINUE
C     CALCULATE NEXT LEG TRAVEL TIME
      J2=1
      DO 270 J1=1,IK2
      J2=J2+1
      K1=KSE(J1,1)
      K2=KSE(J1,2)
      IF (J1.EQ.IK2) GO TO 200
      K3=KSE(J2,1)
      K4=KSE(J2,2)
      GO TO 210
  200 CONTINUE
      K3=IENDP
      K4=1
  210 CONTINUE
      IF (K2.EQ.K4) GO TO 230
      IF (K2.EQ.2) GO TO 220
      CALL TFETCH(KSE(J1,6),PA03XX,K1,K3,DISTOD)
      GO TO 270
  220 CALL TFETCH(KSE(J1,6),PA03XX,K3,K1,DISTOD)
      GO TO 270
  230 IF (K2.EQ.2) GO TO 240
      IF (K1.GT.K3) GO TO 260
```

```
      GO TO 250
  240 IF (K1.LT.K3) GO TO 260
  250 CALL  TFETCH(KSE(J1,6),PA03XX,K1,K3,DISTOO)
      GO TO 270
  260 CALL  TFETCH(KSE(J1,6),PA03XX,K3,K1,DISTOO)
  270 CONTINUE
C     CALCULATE DWELL TIME
      J1=0
  278 J1=J1+1
      IF (J1.GT.IK2) GO TO 300
      I9=KSE(J1,9)
      I10=KSE(J1,10)
      I11=KSE(J1,11)
      IF (KSE(J1,2).EQ.2) GO TO 286
      IF (KSE(J1,6).EQ.0) GO TO 280
      I1=KSE(J1,1)
      CALL FIELD(1,CANDTP(I1,16),KSE(J1,7),17,8)
      GO TO 278
  280 CONTINUE
      J2=J1
  282 J1=J1+1
      IF (KSE(J1,2).EQ.2) GO TO 284
      KSE(J1,7)=0
      I9=I9+KSE(J1,9)
      I10=I10+KSE(J1,10)
      I11=I11+KSE(J1,11)
      IF (KSE(J1,6).EQ.0) GO TO 282
  284 CALL DWELLT(I9,I10,I11,0,0,0,JRDWT,KSE(J2,7),VCADWE,PARM10)
      GO TO 278
  286 IF (KSE(J1,6).EQ.0) GO TO 290
  287 I1=KSE(J1,1)
  288 CALL FIELD(1,CANDTP(I1,16),KSE(J1,7),25,8)
      GO TO 278
  290 CONTINUE
      IF (J1+1.GT.IK2) GO TO 287
      J2=J1
  292 J1=J1+1
      IF (KSE(J1,2).EQ.1) GO TO 294
      KSE(J2,7)=0
      I9=I9+KSE(J1,9)
      I10=I10+KSE(J1,10)
      I11=I11+KSE(J1,11)
      IF (KSE(J1,6).EQ.0) GO TO 292
  294 CALL DWELLT(0,0,0,I9,I10,I11,JRDWT,KSE(J1,7),VCADWE,PARM10)
      GO TO 278
  300 CONTINUE
C     ASSIGN THE SCHEDULED TIME
      KSE(1,8)=ICSBTR(1,8)
      DO 370 J1=2,IK2
      IF (KSE(J1,2).EQ.2) GO TO 340
      IF (KSE(J1-1,2).EQ.2) GO TO 330
      KSE(J1,8)=KSE(J1-1,8)+KSE(J1-1,6)+KSE(J1,7)
      GO TO 370
  330 KSE(J1,8)=KSE(J1-1,8)+KSE(J1-1,6)+KSE(J1-1,7)
     1 +KSE(J1,7)
      GO TO 370
```

```
  340 CONTINUE
      IF (KSE(J1-1,2).EQ.2) GO TO 350
      KSE(J1,8)=KSE(J1-1,8)+KSE(J1-1,6)
      GO TO 370
  350 KSE(J1,8)=KSE(J1-1,8)+KSE(J1-1,6)+KSE(J1-1,7)
  370 CONTINUE
      CALL FESIBL(IK2,IFLAG1,IFLAG5,
     1 CANDTP,DISTOD,DISTOO,IVWHTR,KSE,
     2 IBGNP,IBGNS,PA03XX,IENDP,IENDS,
     3 PARM03,PARM04,PARM05,PARM06 )
      IF (IFLAG1.EQ.0) GO TO 400
      CALL RMTIME(IA,IK2,JK,IFLAG2,
     1 CANDTP,ISE,KSE,IC1,IC2,RTB,RTS,SHTB,SHTS,
     2 PARM03,PARM05 )
  400 CONTINUE
  500 CONTINUE
      RETURN
      END
```

APPENDIX J TEST DATA

This appendix contains data on trip information used in test runs. Trip information are real. The meaning of columns are stated below. Trip data are reproduced as exactly as they were received. The data format and the contents are not necessarily the same as used in the RSDAR.

Column	
1	Patron name
2	Trip number
3	Request time (hhmm)
4	Service code
5	Agency
6	Patron type
7	Number of regular patron
8	Number of wheelchair patron
9	X-coordinate of the origin
10	Y-coordinate of the origin
11	X-coordinate of the destination
12	Y-coordinate of the destination

FERRANTE THELMA	1	615	1	1	2	0	1	15	23	15	21
HERSEY MARCIE	2	645	1	1	2	0	1	11	24	15	21
MERLER MURIEL	3	700	1	1	2	0	1	10	20	17	21
BORKAN JENNIE	4	700	1	1	2	0	1	17	21	16	26
PRUCHA MARY	5	700	1	1	2	0	1	9	14	14	19
HUNTER ROBERT	6	700	1	1	2	0	1	12	18	10	17
SANDS JOHN	7	700	1	1	2	0	1	10	15	9	15
HELM GLORIA	8	700	1	1	2	0	1	16	25	13	22
JIMINEZ ROSE AIMEE	9	730	1	1	2	0	1	12	20	15	21
ALEMAN NIRIC	10	730	1	1	2	0	1	15	21	15	21
VAZQUEZ OSVALDO	11	745	1	1	2	0	1	17	21	15	21
ALVAREZ ARMANDO	12	800	1	1	2	0	1	14	20	10	17
ROJAS MANUAL	13	830	1	1	2	0	1	13	20	18	23
CANTY DENISE	14	830	1	1	2	0	1	14	22	15	21
LEVENSON MILDRED	15	845	1	1	2	0	1	18	23	15	20
WHITE SUSAN	16	900	1	1	2	0	1	12	20	15	20
CLEMENCON HELEN	17	900	1	1	2	0	1	17	21	17	21
RODRIGUEZ GILBERTO	18	1000	1	1	2	0	1	14	20	15	21
SIMAKOW NANCY	19	1000	1	1	2	0	1	15	24	16	26
EISAMAN PAMELA	20	1000	1	1	2	0	1	11	20	11	24
VAZQUEZ OSVALDO	21	1100	1	1	2	0	1	15	21	17	21
ALVARINO JESUS	22	1100	1	1	2	0	1	13	20	11	24
KRIVIT DOROTHY	23	1100	1	1	2	0	1	18	23	18	23
KALMUS IRENE	24	1130	1	1	2	0	1	17	21	18	25
ALEMAN NIRIO	25	1130	1	1	2	0	1	15	21	14	21
LEVENSON MILDRED	26	1130	1	1	2	0	1	15	20	14	24
WHITE SUSAN	27	1130	1	1	2	0	1	15	20	12	20
MAXWELL WILLIE	28	1145	1	1	2	0	1	14	25	15	21
SIMIKON NANCY	29	1145	1	1	2	0	1	16	26	15	24
TURNER LEONARD	30	1200	1	1	2	0	1	15	23	11	24
CLEMENCON HELEN	31	1230	1	1	2	0	1	17	21	17	21
KALMUS IRENE	32	1330	1	1	2	0	1	18	25	17	21
KRESS JOAN	33	1330	1	1	2	0	1	8	20	9	19
HUNTER BEVERLY	34	600	1	1	1	1	0	14	26	15	20
PERKINS JAMES	35	600	1	1	1	1	0	14	27	15	21
PALMER RON	36	600	1	1	1	1	0	14	27	15	20
MCNALLY JOHN	37	600	1	1	1	1	0	16	26	15	20
LEVINE HARRIET	38	600	1	1	1	1	0	16	26	17	21
SUAREZ HILDA	39	600	1	1	1	1	0	14	21	13	18
DE HARVE EDWARD	40	600	1	1	1	1	0	12	24	15	21
GARCIA ALBERT	41	600	1	1	1	1	0	11	19	12	20
WILCOX ELIZABETH	42	600	1	1	1	1	0	15	22	15	21
GORDON JIM	43	600	1	1	1	1	0	14	22	15	21
KENDRICK DENISE	44	600	1	1	1	1	0	14	23	15	22
DAY DANIEL	45	600	1	1	1	1	0	12	20	15	21
MALLIE ELEANOR	46	615	1	1	1	1	0	15	20	15	21
CRUZ ELSA	47	615	1	1	1	1	0	13	23	15	20
FIELDS LINDA	48	615	1	1	1	1	0	13	24	17	23
WHITE AL	49	615	1	1	1	1	0	16	26	14	21
BYRD FRED	50	625	1	1	1	1	0	15	23	17	22
KAZMARK RONALD	51	630	1	1	1	1	0	13	19	15	20
FERGUSON GRANVILLE	52	630	1	1	1	1	0	14	25	15	21
DOWNER RICHARD	53	630	1	1	1	1	0	15	27	16	25
LEVINE ROSS	54	630	1	1	1	1	0	15	27	13	25
GUDWIN FLORA	55	630	1	1	1	1	0	16	25	15	21
WALLIS JEFFREY	56	635	1	1	1	1	0	16	22	15	20

BURRIS CLARK	57	645	1	1	1	1	0	14	19	17	21
WEBSTER GEORGE	58	645	1	1	1	1	0	14	22	15	21
CLETZKY MARSHALL	59	700	1	1	1	1	0	17	25	15	21
BLIER MARK	60	700	1	1	1	1	0	14	20	17	21
KING CAROL	61	700	1	1	1	1	0	17	22	12	27
POTTER RAYMOUND	62	700	1	1	1	1	0	14	26	10	13
BENRUBE MALLARY	63	700	1	1	1	1	0	17	22	12	18
CERNOBORT LINDA	64	700	1	1	1	1	0	16	25	17	19
RENDELL JUDY	65	700	1	1	1	1	0	14	27	11	21
RICE JOEL	66	700	1	1	1	1	0	11	17	10	13
YUBASZ HELENE	67	700	1	1	1	1	0	13	24	15	20
HEMPHILL BETTY	68	700	1	1	1	1	0	10	17	14	22
BLIER ARLYS HALL	69	700	1	1	1	1	0	14	20	15	20
WARDLE VIOLA	70	700	1	1	1	1	0	16	26	15	21
WILKATIS JOE	71	700	1	1	1	1	0	13	19	13	24
BUFKE MARION	72	700	1	1	1	1	0	16	26	14	21
TURRINI LAWRENCE	73	700	1	1	1	1	0	15	21	10	15
NABUTOVSKY BARBARA	74	700	1	1	1	1	0	11	18	10	13
LEEDS LOMA	75	700	1	1	1	1	0	6	9	14	23
LAIDLOW THOMAS	76	705	1	1	1	1	0	13	22	17	19
SHELDON MARTHA	77	705	1	1	1	1	0	16	25	15	21
CANN ELAINE	78	710	1	1	1	1	0	15	23	14	21
MCKENZIE ANNE	79	710	1	1	1	1	0	15	21	15	21
THOMAS ANNELY	80	710	1	1	1	1	0	15	22	15	20
JOHNSON ANN	81	715	1	1	1	1	0	13	25	11	21
SILVERO HAYDEE	82	715	1	1	1	1	0	17	21	15	21
CAMP CHARLOTTE	83	715	1	1	1	1	0	13	21	11	23
GODFREY ROSEANN	84	715	1	1	1	1	0	9	18	17	21
MATHIS GREGORY	85	715	1	1	1	1	0	17	19	17	19
DIAZ SARAH	86	715	1	1	1	1	0	13	24	13	24
RANDOLF ANNIE MAE	87	715	1	1	1	1	0	15	23	14	21
CAMERON GLORIA	88	715	1	1	1	1	0	14	21	15	20
SIPPIO JANICE	89	720	1	1	1	1	0	14	22	15	21
MILLER BRUCE	90	720	1	1	1	1	0	11	20	14	21
LAATSCH JUDY A	91	725	1	1	1	1	0	13	21	15	20
LEVIT IDA	92	730	1	1	1	1	0	16	21	15	20
PRYOR JANET	93	730	1	1	1	1	0	16	22	15	20
ALTMAN RUTH	94	730	1	1	1	1	0	10	18	10	13
CALDERON WILMA	95	730	1	1	1	1	0	13	23	14	24
JAMES MAGGIE	96	730	1	1	1	1	0	14	24	14	22
CONDARES GEORGE	97	730	1	1	1	1	0	14	19	14	21
GREENWALD JAY	98	730	1	1	1	1	0	16	21	15	23
WILLIAMS JOSEPH	99	730	1	1	1	1	0	14	23	15	26
CLEARY ENID	100	730	1	1	1	1	0	12	20	13	24
PHINNEY DONALD	101	730	1	1	1	1	0	13	20	15	20
FINKEL RONALD	102	730	1	1	1	1	0	17	27	14	21
JIMINEZ ROSE	103	730	1	1	1	1	0	12	20	15	21
GILMORE FRANCES	104	730	1	1	1	1	0	14	22	14	24
ZAMORA OTTO	105	730	1	1	1	1	0	8	20	10	16
BLANCO JERONIMO	106	730	1	1	1	1	0	15	20	14	21
HAJ FAREED	107	730	1	1	1	1	0	12	19	12	19
SANCHEZ ANNA L	108	745	1	1	1	1	0	9	14	12	18
DONLEY DALE A	109	745	1	1	1	1	0	15	24	12	26
GRAY LORNA	110	745	1	1	1	1	0	9	14	10	17
FIX COLEEN M	111	745	1	1	1	1	0	16	24	14	24
SANDMAN MILDRED	112	745	1	1	1	1	0	17	22	18	22

GALLARDO PEDRO	113	750	1	1	1	1	0	11	18	10	17
PARSONS RUTH	114	800	1	1	1	1	0	18	23	18	22
GUSMAN AUGUSTINA	115	800	1	1	1	1	0	10	19	15	22
SIONE RITA	116	800	1	1	1	1	0	17	23	14	24
WEBER RUTH	117	800	1	1	1	1	0	8	18	15	20
HERNANDEZ ELENA	118	800	1	1	1	1	0	12	20	15	21
PEREZ SALOMON	119	800	1	1	1	1	0	17	20	15	21
KWASNICKI VICTORIA	120	800	1	1	1	1	0	11	25	9	18
LONDON PATRICIA	121	800	1	1	1	1	0	13	22	15	21
REGGVINTI THERESA	122	800	1	1	1	1	0	14	25	13	25
MELNICK ALBERT	123	810	1	1	1	1	0	11	19	9	18
WINTERS THERESA	124	810	1	1	1	1	0	13	22	9	18
SCHWARTZ SETH	125	815	1	1	1	1	0	18	22	15	21
ZINGARO MARION	126	815	1	1	1	1	0	14	20	15	20
FALDOSKI JENNIE	127	815	1	1	1	1	0	14	27	15	25
ALLEN JAMES F	128	815	1	1	1	1	0	10	17	13	26
GORMAN FRANK	129	815	1	1	1	1	0	15	26	13	20
KIMBRO EOLYN	130	820	1	1	1	1	0	14	24	15	21
DE COSTE	131	825	1	1	1	1	0	11	16	9	18
HARTNETT PATRICIA	132	825	1	1	1	1	0	10	19	9	18
SCHAUR HAROLD	133	825	1	1	1	1	0	9	19	9	18
LEFF BEN	134	830	1	1	1	1	0	16	24	18	22
BUTTA JOSEPH	135	830	1	1	1	1	0	8	18	12	18
SIMON HARRIET	136	830	1	1	1	1	0	13	27	15	20
QUINONES MARIA	137	830	1	1	1	1	0	14	20	16	17
FERNANDEZ JOSE	138	830	1	1	1	1	0	12	20	15	22
RINGER ANN	139	830	1	1	1	1	0	17	25	18	23
BILLINGTON MARK	140	830	1	1	1	1	0	10	18	9	18
SPINELLI WILLIAM	141	830	1	1	1	1	0	16	25	13	26
CALDERON CECILIA	142	830	1	1	1	1	0	9	14	17	25
WEST SUSAN M	143	830	1	1	1	1	0	11	18	9	18
WILSON MARTHA J	144	830	1	1	1	1	0	13	19	13	19
RAMONDAS RASA	145	835	1	1	1	1	0	10	18	9	18
DIRUME SELIO	146	835	1	1	1	1	0	11	18	9	18
YOCHIM ARTHUR	147	840	1	1	1	1	0	10	19	11	19
HUGGINS RONALD	148	840	1	1	1	1	0	9	19	9	18
FIX COLLEEN M	149	845	1	1	1	1	0	16	24	15	20
DAPONTE DALE	150	845	1	1	1	1	0	14	21	15	20
SPERRAZZI LEONARD	151	845	1	1	1	1	0	16	26	15	20
LEON BETSY	152	845	1	1	1	1	0	12	25	13	26
THOMAS RICHARD	153	845	1	1	1	1	0	15	21	15	21
APARDO SOCCRRO	154	900	1	1	1	1	0	15	23	15	21
FEDER ISAAC	155	900	1	1	1	1	0	18	23	17	21
MCCOLLISTER JAMES	156	900	1	1	1	1	0	14	27	16	25
HERRERA MARIA	157	900	1	1	1	1	0	14	22	13	19
MENDIOLA DULCE	158	900	1	1	1	1	0	9	14	14	19
RAMIREZ BEATRICE	159	900	1	1	1	1	0	13	19	15	21
BARNES EDITH	160	900	1	1	1	1	0	15	23	15	20
DUNN ARTHUR	161	900	1	1	1	1	0	13	19	14	21
HARDEN DOROTHY	162	900	1	1	1	1	0	14	19	11	24
MORALES ISHMAEL	163	900	1	1	1	1	0	15	21	15	21
LANTZ MICHAEL	164	900	1	1	1	1	0	12	19	11	16
WALKER NANCY	165	900	1	1	1	1	0	13	27	14	26
DRESSLER DOROTHY	166	900	1	1	1	1	0	12	18	12	17
SCHWARTZ BEATR	167	900	1	1	1	1	0	17	22	17	21
SILVERSTEIN MINA	168	900	1	1	1	1	0	18	22	17	21

AKERMAN CYNTHIA	169	900	1	1	1	1	0	13	23	13	23
DOUGLAS TOM	170	905	1	1	1	1	0	14	19	17	27
BARON FRED	171	915	1	1	1	1	0	17	20	15	20
HEARN BONNIE	172	915	1	1	1	1	0	14	22	12	19
ZAVAC JEFF	173	915	1	1	1	1	0	12	19	14	23
PHANG ANGELA	174	930	1	1	1	1	0	11	20	13	19
REE LEE	175	930	1	1	1	1	0	14	24	14	24
ARODIS ANA M	176	930	1	1	1	1	0	10	20	12	20
MICHAUD DAVID	177	930	1	1	1	1	0	11	25	13	26
KENDRICK AIDA	178	930	1	1	1	1	0	14	24	14	24
MILGROM HELEN	179	930	1	1	1	1	0	11	18	11	15
MILLEY SADIE	180	930	1	1	1	1	0	18	22	18	23
YOUNG PAMELA M	181	930	1	1	1	1	0	15	22	14	24
LOMBARDI MINERVA	182	945	1	1	1	1	0	16	23	14	22
SPRAYGREEN GERTRUDE	183	945	1	1	1	1	0	14	23	17	21
SIEBEL MATHILDA	184	945	1	1	1	1	0	16	26	15	26
CASH RUTH	185	1000	1	1	1	1	0	17	21	15	21
BRYAN JAMES M	186	1000	1	1	1	1	0	17	21	13	21
WASSERMAN FANNIE	187	1000	1	1	1	1	0	16	21	15	26
DALTON CLARA SMITH	188	1000	1	1	1	1	0	11	14	13	19
FRIEH CORA	189	1000	1	1	1	1	0	11	14	10	15
GLOVIN ETHEL	190	1000	1	1	1	1	0	15	20	11	24
LINARES VIOLETTA	191	1000	1	1	1	1	0	13	23	13	23
SAMPEDRO EULALITA	192	1000	1	1	1	1	0	10	19	15	19
BLOMBERG LEE	193	1000	1	1	1	1	0	11	18	13	20
TURNER JOHANN	194	1000	1	1	1	1	0	13	19	10	18
RINGER ANN	195	1000	1	1	1	1	0	18	23	15	20
HOROWITZ ESTELLA	196	1000	1	1	1	1	0	8	18	11	17
SIGMAN SIDNEY	197	1000	1	1	1	1	0	17	23	17	25
CAMPION GRACE	198	1030	1	1	1	1	0	14	21	15	21
CHONSKY ANNA	199	1030	1	1	1	1	0	17	23	13	20
MIKUCIONIS BOBERT	200	1030	1	1	1	1	0	15	25	13	23
BITINER SOPHIE	201	1030	1	1	1	1	0	11	26	15	21
HEALY LILLIE	202	1030	1	1	1	1	0	14	19	13	20
MEYERS EUGENE	203	1030	1	1	1	1	0	14	19	14	21
ALLEN MOE	204	1030	1	1	1	1	0	16	27	16	26
WILLIAMS ROSE MAE	205	1030	1	1	1	1	0	14	22	13	23
GREENFIELD EDITH	206	1045	1	1	1	1	0	16	26	17	22
MEYER ROSE	207	1045	1	1	1	1	0	17	23	11	20
MARGULTES GUSSIE	208	1045	1	1	1	1	0	14	20	15	20
SIEGEL MATHILDA	209	1045	1	1	1	1	0	15	26	16	26
SCHWARTZ SETH	210	1100	1	1	1	1	0	15	21	18	22
CHATLOS RUTH	211	1100	1	1	1	1	0	14	21	15	21
ROSS MARY	212	1100	1	1	1	1	0	11	19	10	17
KRAUSS PEARL	213	1100	1	1	1	1	0	17	27	18	22
WILLIAMS PATTY	214	1100	1	1	1	1	0	14	25	15	26
ROSENTHAL DAISY	215	1100	1	1	1	1	0	15	20	15	20
HARDEN DOROTHY	216	1100	1	1	1	1	0	11	24	14	19
DEANE WANDA	217	1100	1	1	1	1	0	16	26	15	21
DAVIS FANNIE	218	1100	1	1	1	1	0	12	21	17	23
WOODARD ELBERTA	219	1100	1	1	1	1	0	13	20	13	19
LEEDS LOMA	220	1100	1	1	1	1	0	14	23	6	9
NEUBERG LEAH G	221	1100	1	1	1	1	0	18	26	17	24
CAUSO ANA	222	1100	1	1	1	1	0	15	20	14	20
KURTZ ELLIE	223	1100	1	1	1	1	0	15	25	16	25
LEVICK MICHAEL	224	1115	1	1	1	1	0	16	21	14	27

CAPLAN BERNARD	225	1115	1	1	1	1	0	18	23	15	20
BOWLES E C	226	1115	1	1	1	1	0	15	23	15	21
BRYAN JAMES M	227	1130	1	1	1	1	0	13	21	17	21
NEIMAN MORRIS	228	1130	1	1	1	1	0	18	25	17	21
STONE RITA	229	1130	1	1	1	1	0	14	24	17	23
FIX COLLEEN M	230	1130	1	1	1	1	0	15	20	9	14
GRAY LORNA	231	1130	1	1	1	1	0	10	17	9	14
KAUFMAN RUTH	232	1130	1	1	1	1	0	16	24	15	22
ZAMORA OTTO	233	1130	1	1	1	1	0	10	16	8	20
BARNES EDITH	234	1130	1	1	1	1	0	15	20	17	25
MICHAUD DAVID	235	1130	1	1	1	1	0	13	26	11	25
REGGVINTI THERESA	236	1130	1	1	1	1	0	13	25	14	25
CALDERON WILMA	237	1200	1	1	1	1	0	14	24	13	23
HERRERRA MARIA	238	1200	1	1	1	1	0	13	19	14	22
CHARBONNEAU WILFRED	239	1200	1	1	1	1	0	15	19	15	21
BARNES EDITH	240	1200	1	1	1	1	0	15	23	17	25
MEYER ROSE	241	1200	1	1	1	1	0	11	20	17	23
KEE LEE	242	1200	1	1	1	1	0	14	24	14	24
TURNER JOHANN	243	1200	1	1	1	1	0	10	18	13	19
HOROWITZ ESTELLE	244	1200	1	1	1	1	0	11	17	8	18
ZLOCKOWER BERNARD	245	1200	1	1	1	1	0	15	26	15	21
RINGER ANN	246	1200	1	1	1	1	0	15	20	17	25
WALKER NANCY	247	1200	1	1	1	1	0	14	26	13	27
RICK SARA	248	1200	1	1	1	1	0	16	21	17	21
PEREZ SALOMON	249	1200	1	1	1	1	0	15	21	13	21
DAVIS EUGENE H	250	1200	1	1	1	1	0	14	21	14	20
POND FRANK	251	1215	1	1	1	1	0	9	19	13	21
MCKENSIE ANNE	252	1215	1	1	1	1	0	15	21	15	24
WEBER RUTH	253	1215	1	1	1	1	0	15	20	15	23
ARFISKY IDA	254	1215	1	1	1	1	0	18	23	17	21
WOODWARD ELBERTA	255	1215	1	1	1	1	0	13	19	13	20
HIRSCH DEBORAH	256	1230	1	1	1	1	0	18	23	17	21
FEDER ISAAC	257	1230	1	1	1	1	0	17	21	18	23
GARFUNDEL HELEN	258	1230	1	1	1	1	0	14	20	15	24
GORDON DORA	259	1230	1	1	1	1	0	17	21	16	26
LOMBARDI MINERVA	260	1230	1	1	1	1	0	14	22	16	23
WELLS GLADYS	261	1230	1	1	1	1	0	18	26	15	20
WILLIAMS PATTY	262	1230	1	1	1	1	0	15	26	14	25
CASTRO AMA	263	1230	1	1	1	1	0	11	24	17	25
SHAVER ANDREW	264	1230	1	1	1	1	0	12	20	10	17
CARLYSLE MAE	265	1230	1	1	1	1	0	16	22	14	21
ALLEN MOE	266	1230	1	1	1	1	0	16	26	16	27
SIGMAN SIDNEY	267	1230	1	1	1	1	0	17	25	17	23
GONZOLAS ROSA	268	1230	1	1	1	1	0	13	21	15	21
SCHWARTZCACH BEAT	269	1230	1	1	1	1	0	17	21	17	22
PARROTTA ANNA	270	1230	1	1	1	1	0	18	26	18	26
SPRAYGREEN GERTRUDE	271	1245	1	1	1	1	0	17	21	14	23
LLOYD FLORENCE M	272	1300	1	1	1	1	0	17	21	15	20
SILVERO HAYDEE LOPE	273	1300	1	1	1	1	0	15	21	17	21
LOPEZ MARTIN	274	1300	1	1	1	1	0	15	20	15	23
WEBSTER GEORGE	275	1300	1	1	1	1	0	15	21	14	22
MORALES ISHMAEL	276	1300	1	1	1	1	0	15	21	15	21
GORDON JIM	277	1300	1	1	1	1	0	15	21	14	22
HOROWITZ CLARA	278	1300	1	1	1	1	0	16	25	16	26
PROJANSKY SARAH	279	1300	1	1	1	1	0	16	26	16	27
LONDON PATRICIA	280	1300	1	1	1	1	0	15	21	13	22

BYRD FRED	281	1300	1	1	1	1	0	17	22	15	23
CALDERON CECILIA	282	1300	1	1	1	1	0	17	25	9	14
KURTZ ELLIE	283	1300	1	1	1	1	0	16	25	15	25
MARTELL LOIS	284	1300	1	1	1	1	0	12	25	11	25
LEACH JANICE	285	1315	1	1	1	1	0	17	21	16	24
WILLIAMS JOSEPH	286	1315	1	1	1	1	0	15	26	14	23
ZAVAC JEFF	287	1315	1	1	1	1	0	14	23	12	19
EATON RUTH	288	1315	1	1	1	1	0	10	17	8	11
MOTES ETTA	289	1330	1	1	1	1	0	12	21	12	18
BLY PATSY	290	1330	1	1	1	1	0	15	24	15	21
DIELH TILLIE	291	1330	1	1	1	1	0	12	19	15	21
ALTMAN RUTH R	292	1330	1	1	1	1	0	14	19	10	18
CASTELLANOS ARCHIE	293	1330	1	1	1	1	0	9	19	14	20
GILMORE FRANCIS	294	1330	1	1	1	1	0	14	24	14	22
SAMET WILLIAM	295	1330	1	1	1	1	0	8	12	13	21
SAMET LILLIAN	296	1330	1	1	1	1	0	8	12	13	21
SANDMAN MILDRED	297	1330	1	1	1	1	0	18	22	17	25
QUINONES MARIA	298	1330	1	1	1	1	0	14	21	14	20
SPINELLI WILLIAM	299	1330	1	1	1	1	0	13	26	16	25
JESSEL MARGARET	300	1345	1	1	1	1	0	15	21	13	19

FEDERAL INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary date (Yr. Mo. Day): 8 5 0 2 0 5

02. Summary prepared by *(Name and Phone)*: Howard K. Hung (301) 921-3855

03. Summary action: New [X] Replacement [ ] Deletion [ ]

Previous Internal Software ID

04. Software date (Yr. Mo. Day): 8 5 0 2 0 5

05. Software title: Routing and Scheduling Dial-A-Ride (RSDAR)

06. Short title

07. Internal Software ID

08. Software type
- [ ] Automated Data System
- [X] Computer Program
- [ ] Subroutine/Module

09. Processing mode
- [ ] Interactive
- [X] Batch
- [ ] Combination

10. Application area

General:
- [ ] Computer Systems Support/Utility
- [X] Scientific/Engineering
- [ ] Bibliographic/Textual
- [X] Management/Business
- [ ] Process Control
- [ ] Other

Specific: Dial-A-Ride

11. Submitting organization and address

Center for Applied Mathematics
National Bureau of Standards
Gaithersburg, MD 20899

12. Technical contact(s) and phone

Howard K. Hung
William G. Hall
Robert E. Chapman
(301) 921-3855

13. Narrative

The RSDAR is a heuristic algorithm. It assigns patrons to form subtours in time intervals, and subtours are linked to become a tour. Patrons are chosen to be included in a subtour on the basis of the best remaining time of the base trip. Subtours are selected to be included in a tour on the basis of the best productivity measure.

14. Keywords

Dial-a-ride; heuristic algorithm; routing; scheduling

15. Computer manuf'r and model

16. Computer operating system: UNIX

17. Programing language(s): FORTRAN 77

18. Number of source program statements: Appx 1400

19. Computer memory requirements: 512K Bytes

20. Tape drives

21. Disk/Drum units: 15 MB Winchester Disk Drive

22. Terminals

23. Other operational requirements

24. Software availability: Available [X] Limited [ ] In-house only [ ]

25. Documentation availability: Available [X] Inadequate [ ] In-house only [ ]

26. FOR SUBMITTING ORGANIZATION USE

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO.	2. Performing Organ. Report No.	3. Publication Date
	NBSIR 85-3178		JULY 1985

4. TITLE AND SUBTITLE

Paratransit Advanced Routing and Scheduling System Documentation: Routing and Scheduling Dial-A-Ride Subsystem

5. AUTHOR(S) Howard K. Hung, William G. Hall, Robert E. Chapman

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

U.S. Department of Transportation
Urban Mass Transportation Administration
400 7th Street S.W.
Washington, D.C. 20390

10. SUPPLEMENTARY NOTES

[X] Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

The Advanced Routing and Scheduling System (ARSS) is a software system designed to route and schedule patrons in a dial-a-ride environment. The system consists of three subsystems: CONENV, a preprocessor which constructs physical and policy environments; RSDAR, which routes and schedules patrons; and GREPOR, which generates hard copy of all necessary reports. This report provides a description of RSDAR.

The RSDAR is a heuristic algorithm. It assigns patrons to form subtours in time intervals, and these subtours are linked to become a tour.

Patrons are chosen to be included in a subtour on the basis of the best remaining time of the base trip. Subtours are selected to be included in a tour on the basis of the best productivity measure.

The model is written in FORTRAN and complies with the American National Standards Institute X3.9-1978 standard for that language.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

Dial-a-ride; heuristic algorithm; routing; scheduling

13. AVAILABILITY

[X] Unlimited
[ ] For Official Distribution. Do Not Release to NTIS
[ ] Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
[X] Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

136

15. Price

$14.50