A SEARCH AND RESCUE SIMULATION MODEL FOR THE UNITED STATES COAST GUARD

Appendix A

FLOW CHARTS FOR PROGRAMMER LEVEL DOCUMENTATION
The National Bureau of Standards\(^1\) was established by an act of Congress March 3, 1901. Today, in addition to serving as the Nation’s central measurement laboratory, the Bureau is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To this end the Bureau conducts research and provides central national services in four broad program areas. These are: (1) basic measurements and standards, (2) materials measurements and standards, (3) technological measurements and standards, and (4) transfer of technology. The Bureau comprises the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Center for Radiation Research, the Center for Computer Sciences and Technology, and the Office for Information Programs.

**THE INSTITUTE FOR BASIC STANDARDS** provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation’s scientific community, industry, and commerce. The Institute consists of an Office of Measurement Services and the following technical divisions:

Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic and Molecular Physics—Radio Physics\(^2\)—Radio Engineering\(^2\)—Time and Frequency\(^2\)—Astrophysics\(^2\)—Cryogenics\(^2\).

**THE INSTITUTE FOR MATERIALS RESEARCH** conducts materials research leading to improved methods of measurement standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; develops, produces, and distributes standard reference materials; relates the physical and chemical properties of materials to their behavior and their interaction with their environments; and provides advisory and research services to other Government agencies. The Institute consists of an Office of Standard Reference Materials and the following divisions:


**THE INSTITUTE FOR APPLIED TECHNOLOGY** provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations in the development of technological standards, and test methodologies; and provides advisory and research services for Federal, state, and local government agencies. The Institute consists of the following technical divisions and offices:


**THE CENTER FOR RADIATION RESEARCH** engages in research, measurement, and application of radiation to the solution of Bureau mission problems and the problems of other agencies and institutions. The Center consists of the following divisions:


**THE CENTER FOR COMPUTER SCIENCES AND TECHNOLOGY** conducts research and provides technical services designed to aid Government agencies in the selection, acquisition, and effective use of automatic data processing equipment; and serves as the principal focus for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Center consists of the following offices and divisions:


**THE OFFICE FOR INFORMATION PROGRAMS** promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System, and provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:


---

1. Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.
2. Located at Boulder, Colorado 80302.
3. Located at 5295 Port Royal Road, Springfield, Virginia 22151.
A SEARCH AND RESCUE SIMULATION MODEL FOR THE UNITED STATES COAST GUARD

Appendix A

FLOW CHARTS FOR PROGRAMMER LEVEL DOCUMENTATION
This volume is one of a series which documents a Search and Rescue Simulation Model for the United States Coast Guard. The material reported in this documentation was developed by an interdisciplinary team at the National Bureau of Standards with representation from the U.S. Coast Guard under MIPR Z-70099-0-01935.

The complete documentation is comprised of the following:

Volume I  Executive Level Documentation
Volume II  Analyst Level Documentation
Volume III  Programmer Level Documentation for "PREPROCESSOR"
Volume IV  Programmer Level Documentation for "OPSIM"
Volume V  Programmer Level Documentation for "POSTPROCESSOR"
Appendix A  Flow Charts for Programmer Level Documentation
Appendix B  Program Listings for Programmer Level Documentation

The study was initially conducted under the supervision of Martin J. Aronoff; subsequently efforts were supervised by Richard T. Penn, Jr. Technical Project Leadership was supplied throughout the project by Stephen S. Karp. Other participants from the National Bureau of Standards Technical Analysis Division included the following:

Susan S. Chamberlin  Elizabeth E. Leyendecker
Linda K. Cummings  Marcia D. Maltese
Mary Jane Duberg  Patsy L.B. Saunders*
William Elliott, III  Wayne A. Steele
Walter G. Leight  Michael R. Vogt
Joel Levy  Arnold L. Weber
Valuable advice was received from Alan J. Goldman* and Prof. Gustave J. Rath of Northwestern University.

U.S. Coast Guard participants included:

Paul D'Zmura  Gerald L. Underwood
Thomas T. Matteson  Robert R. Wells

Support services were furnished by the following members of the NBS Technical Analysis Division:

Mary M. Abbott  Frances E. Jones
Theresa I. Conrad  Lucinda I. Farrell

* Staff members of the NBS Applied Mathematics Division
# TABLE OF CONTENTS

## I. PREPROCESSOR

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCP</td>
<td>1</td>
</tr>
<tr>
<td>NEEDS</td>
<td>9</td>
</tr>
<tr>
<td>NUCASE</td>
<td>11</td>
</tr>
<tr>
<td>READ</td>
<td>15</td>
</tr>
<tr>
<td>FIELD</td>
<td>17</td>
</tr>
<tr>
<td>DEMGEN</td>
<td>19</td>
</tr>
<tr>
<td>HIST</td>
<td>24</td>
</tr>
<tr>
<td>SELECT</td>
<td>27</td>
</tr>
</tbody>
</table>

## II. OPSIM

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>28</td>
</tr>
<tr>
<td>NUCRU</td>
<td>2</td>
</tr>
<tr>
<td>OPSIM</td>
<td>31</td>
</tr>
<tr>
<td>SRAS</td>
<td>33</td>
</tr>
<tr>
<td>CRES</td>
<td>34</td>
</tr>
<tr>
<td>VEC</td>
<td>37</td>
</tr>
<tr>
<td>OSET</td>
<td>39</td>
</tr>
<tr>
<td>RESAP</td>
<td>40</td>
</tr>
<tr>
<td>ROCA</td>
<td>46</td>
</tr>
<tr>
<td>ROCA</td>
<td>46</td>
</tr>
<tr>
<td>MRAS</td>
<td>47</td>
</tr>
<tr>
<td>NOTIF</td>
<td>48</td>
</tr>
<tr>
<td>TOW</td>
<td>49</td>
</tr>
<tr>
<td>DTD</td>
<td>50</td>
</tr>
<tr>
<td>SRCH</td>
<td>51</td>
</tr>
<tr>
<td>NOTE</td>
<td>page 53</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------</td>
</tr>
<tr>
<td>SASS</td>
<td>page 54</td>
</tr>
<tr>
<td>SSS</td>
<td>page 56</td>
</tr>
<tr>
<td>READY.</td>
<td>page 60</td>
</tr>
<tr>
<td>ARSCH</td>
<td>page 61</td>
</tr>
<tr>
<td>COMPL</td>
<td>page 62</td>
</tr>
<tr>
<td>SSET.</td>
<td>page 63</td>
</tr>
<tr>
<td>FUEL</td>
<td>page 64</td>
</tr>
<tr>
<td>HOMEF.</td>
<td>page 65</td>
</tr>
<tr>
<td>SNDBK.</td>
<td>page 66</td>
</tr>
<tr>
<td>SRISE.</td>
<td>page 68</td>
</tr>
<tr>
<td>XSET</td>
<td>page 69</td>
</tr>
<tr>
<td>SERVE.</td>
<td>page 70</td>
</tr>
<tr>
<td>DELAY.</td>
<td>page 72</td>
</tr>
<tr>
<td>ARVSN.</td>
<td>page 73</td>
</tr>
<tr>
<td>STATS.</td>
<td>page 76</td>
</tr>
<tr>
<td>SRCHEF.</td>
<td>page 77</td>
</tr>
<tr>
<td>ONSCN.</td>
<td>page 78</td>
</tr>
<tr>
<td>COVER.</td>
<td>page 80</td>
</tr>
<tr>
<td>CHEKN.</td>
<td>page 81</td>
</tr>
<tr>
<td>RETN</td>
<td>page 82</td>
</tr>
<tr>
<td>TERM</td>
<td>page 83</td>
</tr>
<tr>
<td>HOME</td>
<td>page 85</td>
</tr>
<tr>
<td>SAQ</td>
<td>page 86</td>
</tr>
<tr>
<td>QUEUE.</td>
<td>page 88</td>
</tr>
<tr>
<td>EXQ</td>
<td>page 90</td>
</tr>
</tbody>
</table>
STINBY ................................................. 94
SVQUE ............................................... 98
WRECK ............................................... 99
ENDSIM. ............................................. 100
DRIVE ............................................... 104
JUMPER. ............................................. 105
Flow Diagram of PCP (Main program)

READ IN C1:30 OFFAC
read in location data for current district
read in search speeds
read in needs matrix
read in district centroids & probabilities
read in opfac location data

CALL NUCASE (NUCASE calls READ, FIELD)

CMAX=#C-cards
NO(I)=#D-cards
for each C-card
convert A3 to integer

I=1,...,CMAX
J=1,...,NO(I)

1
X=XPT
Y=YPT

A

C7A(I),C7B(I) =
or 9999

C130 case?

C130 case?

Yes

Yes

No

No

find distance R to each
distance centroid

assign A1A(I) by Monte Carlo

X_A1A(I)’s
centroid

assign A1A(I) by Minimum R

X=((LONG-C7B)*ZBETA)/60
Y=C7A-LAT

B

C

last C-card

Yes

Yes

No

No

No

Yes

A

A1B(I)=0

No

X=XX
Y=YY

B

case location on
any C-card?

is A1A(I) =DIST ?

location data for
A1B(I)?

X=XPT
Y=YPT

B

X,Y=XDLETA,YDLETA
from A1B(I)
Flow Diagram of PCP
(Main program)

READ IN
C130 OFFAC

read in
location
data for
current district

read in
search
speeds

read in
needs
matrix

read in
district
centroids & probabilities

read in
opfac
location
data

Read in
case

Call NUCASE
(NUCASE calls
READ, FIELD)

PCP
CASE
FILE

\[ \text{CMAX} = \# \text{C-cards} \]
\[ \text{NO}(1) = \# \text{D-cards} \]
for each C-card

convert A3 to integer

\[ \text{I}=1, \ldots, \text{CMAX} \]
\[ \text{J}=1, \ldots, \text{NO}(1) \]
\[ X = \frac{(LONG-C7B) \times \text{BETA}}{60} \]

\[ Y = C7A - \text{LAT} \]

\[ XLOW \leq x \leq XLMT \quad \text{and} \quad XLOW \leq x \leq XLOW \]

\[ X = 2 \quad Y = 2 \]

\[ X = X \quad Y = Y \]

\[ \text{MINCI} = C1(1) \]

\[ D6\text{TOT} = \sum D6(I, J) \]

\[ TSM = \sum (D6(I, J) \times SOA3(D1(I, J))) \]

find MILIES from B16

find NSEV from B17B

\[ B \]

\[ C \]

\[ 3 \]
THIS MAY BE A USER CINIVAL

3

D6TOT>5

Yes → E

No

S1=0
S2=0

1st

D6(I,J) >0

Yes

S2=1

No

D6(I,J) >0

Yes

S2=2

No

S2=0

D

D4FRST(I,J)

D4(I,J) +
D6(I,J)

N=N+1

set NEED(N)

Call NEEDS

4
I & EN£fZAL
ASS
1ST
A
wee
rendered
gy
•10
£fc
QoiR.es
ec
sc
c
st

14: GENERAL ASSISTANCE RENDERED BY SURFACE OR AIR RESOURCE

15: TOW; DEWATER & TOW; SURFACE ESCORT; REFLOATED & TOWED

SURFACE RESOURCE STOOD BY BECAUSE IT CANNOT ESCORT AIRCRAFT CLIENT

-19-
SUB-Routine READ

Convert Status Word Index to Test Previous Read:
IT = 3 IS

Convert Beginning Buffer Index to Test Previous Read:
JF = A58 IF

Calculate Beginning Buffer Index Correction Factor:
KF = IF IF - 2

Status of Previous Read ≥ 0

-1 Status of Previous Read Value ≤ 2

Release Unit Backspace One Record, Attempt Re-Read Up to 3 Times

Status of Read
-0 No. of Words Transmitted
-1 Transmission Not Complete
-2 End of File
-3 Device Error
-4 Transmission Aborted

Note: If status = 1, channel is busy and testing continues until one of the other conditions results

Status of Reread ≥ 0

-1 Status of Reread Value ≤ 2

Stop

-1 Reti
TEST FOR NEW OFPAC, NPREC=0

YES

SAVE OFPAC HEADER: HDR(I), I=1,14

SET NPREC = NO. OF PHYSICAL RECORDS IN OFPAC

INITIALIZE MU AND C130 FLAGS:
MFLAG = 0,
CFLAG = 0

CFLAG = 1

TEST TYPE OF HEADER

C130

MU

MFLAG = 1

OPFAC

INCREMENT OFPAC RECORD COUNTER: NPREC = NPREC + 1

READ RECORD INTO LOAD(1F)

CSET I$=I$, I$=J$
LMT = 70P - LMT
NEXT READ

NO
ENTRY FIELDS

D-CARD ZERO FILLED (CIISO CASES ONLY)

Determines A/N OR INTEGER EQUIVALENT OF D(I,J), D0(I,J), D1(I,J), D2(I,J), D3(I,J), D4(I,J), D5(I,J), D6(I,J)

J = J - 1

Extracts information pertaining to each D-CARD WHERE I = 1, ..., CMAX AND J = 1, ..., NO(I)

D4(I,J): D3(I,J)

RET

RET
EXOCENOUS EVENT START

Cross check number of resources in initialization data

Print error message

NSGA 1 <= I <= R

Determine correct shift for start of simulation

Set resource attributes - STN, TYPE, XR, YR

Set PSHT to the first shift of a week end day, holiday

Is the first day a holiday?

Calculate LIMIT

Create and cause the first rise

Create and cause the first set

Return to MAINXX

#MAINXX is the timing routine
A
Call CRES
Call VEC
Call OSET

B
Call CRES
Call VEC
Call OSET

C
Call SASS

D
Is there a capable idle resource with an available crew to serve the need?
Yes

D
Is there a capable idle resource with an available crew to serve the need?
No

D
Is there an available crew to serve the need?
Yes

D
Is the resource a small boat or cutter?
Yes

D
Update TOQUE and TINT
Set RSRG=IR
Set SASG=1
Remove need from QQUE
Call SSS
SUBROUTINE OPSIM

Create CASE

- Read CASE attributes

- Set STATN(CASE) = 0
  - yes: set STATN(CASE) to closest station in district with PSTN = 1
  - no: set STATN(CASE) to converted station number

- Does STATN(CASE) exist?
  - yes: Set STATN(CASE) equal to the closest station of (P*A) with PSTN=1
  - no: PSTN(STATN(CASE))

- Does (P*A)-set of stations including the primary and all of its adjacents

C:

- Is this a long search CASE only?
  - yes: Read NEED(CASE) and OST(CASE) of single resource CASE
  - no: Does CASE have tow needs only?
    - yes: yes to A
    - no: no to C

A:

- for each non-tow need
  - Create NOTIF
  - Initialize NOTIF attributes
  - File NOTIF in NSET(CASE)
  - LOOP}

D:

- 31
DO for each \( \text{NOTIF} \) in \( \text{NSET(CASE)} \)

Read \( \text{NEED(NOTIF)} \)

\( \text{OST(NOTIF)} \) and \( \text{DELTA(NOTIF)} \)

\( \text{LOOP} \)

Does multi-resource CASE have a key need?

yes

Read \( \text{ITOW(CASE)} \)

Is multi-resource CASE an air escort?

no

\( \text{A} \)

no

\( \text{C} \)

\( \text{no} \)

Are there any needed other than air escort or search needs?

\( \text{yes} \)

Is \( \text{COUNT-\text{LIMIT}} \) ?

\( \text{no} \)

Is there an impossible case attributes?

\( \text{yes} \)

Is this a single resource CASE?

\( \text{no} \)

Is this a long search CASE?

\( \text{yes} \)

\( \text{Return to \( \text{MAINXX} \)} \)

Call \( \text{SSAS(CASE)} \)

\( \text{Call \( \text{SRCH(CASE)} \)} \)

Empty \( \text{NSET(CASE)} \)

\( \text{File \( \text{CASE into EXCS} \)} \)

\( \text{Destroy \( \text{CASE} \)} \)

\( \text{Return to \( \text{MAINXX} \)} \)

\( \text{File \( \text{CASE EXCS} \)} \)

\( \text{Return to \( \text{MAINXX} \)} \)

Return to \( \text{MAINXX} \)

\( \text{Return to \( \text{MAINXX} \)} \)
Subroutine SRAS

CALL CRES

II = 0 ?

no

KNT = 0 ?

no

set ITOL = 3

Increase FAIL1 by 1

File CASE in EXCS

Return to calling program

Value of II is returned from CRES; II=0, no capable resource types; II=1, capable resource types do exist

yes

Value of KNT is returned from CRES; KNT=0, no capable resources; KNT > 0, KNT capable resources are in QUE

set ITOL = 4

Increase FAIL2 by 1

File CASE in EXCS

Return to calling program

CALL VEC

CALL RESAP

ires = 0 ?

yes

CALL OSET

CALL SERVE

Value of IRES is returned from RESAP,
IRES = 0, no available resource to serve CASE
IRES ≠ 0, resource that will serve the CASE

Return to calling program

Is NEED an air escort?

yes

Set TINO,REA, SIGNAL; Increase NQUE by 1

File CASE in QUE

Return to calling program

no

Set ITOL=5

File CASE in EXCS

Return to calling program
CRES Cont. (2)

Set IRAY2(1) = STATN(CASE)

IF, FOR I = (1) (NACS(STATN(CASE)))

ACS(STATN(CASE),1) = 0

yes

J = J + 1
IRAY2(J) = ACS(STATN(CASE),1)

L O O P

no

IRAY1 now contains all capable resource

CUT(STATN(CASE),1) = 0

yes

J = J + 1
IRAY2(J) = CUT(STATN(CASE),1)

L O O P

no

IRAY2 now contains all stations capable of serving the single resource CASE

Value of RAP

RAP = 4.5

-35-

K = 0

DO, FOR I = (1) (NRST)

Store AND(IAND,MASK(I)) in ITEMP

ITEMP > 0

yes

K = K + 1
IRAY1(K) = 1

L O O P

no

IRAY1 now contains all capable resource

SIS(CASE) = 0

yes

C

no

This a multi-resource CASE

no

C

C + 1, 2, 3, 6

RAP = 4.5

-35-

POB(CASE) > 5

no

IRAY1(1) = 16

yes

POB(CASE) > 10

no

IRAY1(1) = 24

yes

POB(CASE) > 18

no

IRAY1(1) = 25

yes

POB(CASE) = 25

no

IRAY1(1) = 26

yes

IRAY1(1) = 27

Return to Calling Program

IAND = CAP(IRAY1(1))

DO, FOR I = (1) (NRST)

Store AND(IAND,MASK(I)) in ITEMP

K = 0

L O O P

yes

IAND > 0

no

KONTR = 0

II = 0

Return to Calling Program

0

yes

IRAY1 now contains all capable resource
IPAY2 now contains all stations capable of serving the CASE.
Subroutine VEC

Resource is at its home station

Resource is moving

D is the distance between the resource and its destination

TVE = D/SoA1

TVE = D/SoA2

C = (TIME-DEP) * SoA1

C = (TIME-DEP) * SoA2

C = D

C = D

X = XR - XC
Y = YR - YC

D = \sqrt{X^2 + Y^2}

SHELL > SLIM?

SHELL > SLIM?

C = 0.1
VEC Cont.

\[ \begin{align*}
X & = X_r - \frac{(C_X)}{D} \\
Y & = Y_r - \frac{(C_Y)}{D} \\
X & = X_r - X_c \\
Y & = Y_r - Y_c \\
D & = \sqrt{X^2 + Y^2} \\
\text{SWELL(CASE)} & > \text{SLIM} \\
\text{TVEC} & = \frac{D}{SOA_1} \\
\text{TVEC} & = \frac{D}{SOA_2} \\
\text{LOOP} \\
\text{RETURN to calling program}
\end{align*} \]
Subroutine RESAP

VALUE of IRAP?
IRAP = 3, 4, 5

Search for an idle resource at the primary station
IRAP = 1, 2, 6

DO, FOR EACH resource in RME

IB ≠ 0

Does station of resource = primary station of CASE?

Does station of resource = any of the aircraft covering stations of the primary?

Does station of resource = any of the cutters of the primary?

STN(IRS) = STATN(CASE)

STN(IRS) = ACS(STATN(CASE))

STN(CASE) = CUT(STATN(CASE))

EIAT = 0, 0

IVAL = 2

Call ROCA

Set JRS = IRS

Value of IRAP

300
IRAP = 1, 6

400
IRAP = 3, 4, 5

900
IRAP = 2

LOOP

-40-
Search for an idle resource in RQNE

DO, FOR EACH resource in RQNE

IF IB ≠ 0
  yes
  Loop

ELSE
  no
  EIAT ≠ 0.0
    yes
    Call ROCA
  no
  no
  Call ROCA

Check CREW ≥ 1

Set JRS = IRS

Return to calling program

Value of IRAP

IRAP = 3, 4
IRAP = 1, 2, 6
IRAP = 5

Return to calling program

Set JRS = IRS
Search for an idle resource at the adjacent stations.

Are there any adjacent stations? yes

DO, FOR EACH resource IN QUEUE

IB ≠ 0 yes

STN(IRS) = ADJS(STATN(CASE)) no

STN(IRS) = ACS(ADJS(STATN(CASE))) yes

ETAT = 0.0 yes

CALL ROCA no

IVAL = 2 yes

LOOP

CALL ROCA no

Set JRS=IRS

Return to drilling program

Set JRS=IRS

VALUE OF IRAP

IRAP=1

IRAP=2

IRAP=3,4,5

IRAP=6

IRAP=800

IRAP=900

RESAP Cont.
Search for a resource that can be interrupted in QUEUE.

DO, FOR EACH resource IN QUEUE

IB = 0

if IB = 0

HCREW = 0

if HCREW = 0

PRIOR(IRS) ≥ PRI(CASE)

if PRIOR(IRS) ≥ PRI(CASE)

IDUM = 100

X = TVEC(IRS)

Y = COST(IRS)

Z = WVEC(IRS)

CALL S40(IRS)

LSTR = 1

LABEL 2

return

else

PRIOR(IRS) < IDUM

if PRIOR(IRS) < IDUM

IDUM = PRIOR(IRS)

JRS = IRS

return

else

end LOOP

end else

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if

end if
Subroutine ROCA

- Is resource operational?
  - yes: K=SHIFT-BCREW
  - no: Set II=2

- Is there an available crew?
  - yes: Set II=1
  - no: Increase BCREW by 1

- K < CL
  - yes: Set II=2
  - no: Increase BCREW by 1

- SYTAG(CASE) = 1
  - yes: Return calling program
  - no: Create STNBY

- Increase both NSTBY and TOSBY by 1
  - yes: Cause STNBY at TIME + 2 hours
  - no: Return to calling program
SUBROUTINE MRAS (CASE)

\( \text{NNN(CASE)} = 0 \)

\( \text{STOS = 0} \)
\( \text{XTOS = 0} \)

DO, for each NOTIF in NSET (CASE)

\( \text{OST(NOTIF)} > \text{XTOS} \)

\( \text{XTOS = OST(NOTIF)} \)

\( \text{STOS = STOS * OST(NOTIF)} \)

LOOP

\( \text{TE = (1 - GAMMA(CASE))} \)
\( \text{XTOS * GAMMA(CASE) * STOS} \)

DO, for each NOTIF in NSET (CASE)

Cause NOTIF at
\( \text{DELTA(NOTIF)} \)
\( \text{TE - OST(NOTIF)} \)
\( \text{TIME NUMBER(NOTIF) = 0} \)
\( \text{COMP(NOTIF) = 0} \)

LOOP

\( \text{NNN(CASE)} = 0 \)

no

CALL TON (CASE)

yes

Return to calling routine
ENDOGENOUS EVENT NOTIF

JRAP = RAP
IF RAP=4, JRAP=2
IF RAP=5, JRAP=6
CALL CRES

II = 0
yes
FAIL1(STAIN) = FAIL1 + 1
ITOL(CAS(NOTIF)) = 5

no

KONTR = 0
yes
FAIL2(STAIN) = FAIL2 + 1
ITOL(CAS(NOTIF)) = 4

no

KONTR > 1
no

DO, FOR EACH IRS OF ROPE

IB(IRS) ≤ 1
no

MFLG ≠ 9
no

ACASE/CAS(NOTIF)

yes

ACASE/CAS(NOTIF)

yes

LOOP

KRES(NOTIF) = IRES

TRES=0
yes
CALL QUEUE (NOTIF, M)

no

COMP(NOTIF)=2
CALL SERVE (CAS, NOTIF, IRES)

no

Return to MAINXX

Execute loop when there is only one resource in the district that can serve the need; Is that resource presently covering only on the case; if yes, let it serve the incoming need.
SUBROUTINE TOW (CASE)

CALL DTD(CASE, XD, YD, D), GAP = MIN(HD, D/2.), M = MAX(CASE)

DO, for I = (1) (M)

CREATE NOTIF
FILE NOTIF IN
NSET (CASE)

I = M or
D < 0.0

yes

XHAND = XD
YHAND = YD

no

YHAND = YD + GAP*,
(YC - YD)/D
XHAND = XD + GAP*,
(XC - XD)/D

NNN > 0
or I = 2

no

COMP (NOTIF) = 0
CAUSE NOTIF AT
TIME

yes

LOOP

Return to calling routine
SUBROUTINE DTD(CASE, XD, YD, D)

NPATRL < 0

DO, for I = (1) (NPATRL)

STATN(CASE) = IPAT(I)

XD = XS(STATN)

YES

XD = XS(ADJS(STATN, 1))

NO

YD = YS(ADJS(STATN, 1))

LOOP

XD = XS(STATN)

YES

YO = YS(STATN)

NO

D = SORT ((XC - XD)**2 + (YC - YD)**2)

Return to calling routine
DO FOR I=1 SIS(CASE)

Create NOTE

SM(NOTE)=
PRISM(SIS(CASE),
I) * TSN(CASE)

Set other NOTE attributes

File NOTE in SIS(CASE)

LOOP

TIME< (SUNRISE
-XR#)

no

TIME< SUNSET

yes

K = 0

Night Time

DO, FOR EACH I in SIS(CASE)

K # 0 yes

no

CALL SASS

LJK # 0 yes

B

no

JRAP = RAP

RAP = 4 yes

JRAP = 2

no

RAP = 5 yes

JRAP = 4

no

CALL RESAP

IR = 0 yes

Set SFLAG, SAS;}

no

CALL SSS

File I in LIST

L = K + 1

LOOP

Return to calling program

Subroutine SRCH

K = 0

Day Time

DO, FOR EACH I in SIS(CASE)

K # 0 yes

no

CASE called I at TIME+EPSLN

CALL SASS

LJK # 0 yes

B

no

JRAP = S1

RAP = 4 yes

JRAP = 2

no

RAP = 6 yes

JRAP = 4

no

CALL RESAP

IR = 0 yes

Set SFLAG, SAS;

no

File I in LIST

K = K + 1

LOOP

Call SSS
SRCH Cont.

-52-
ENDOGENOUS EVENT NOTE

- If SUNSET-TIME > SUNRISE, then:
  - Neither CALL SASS nor CALL RESAP
  - File NOTE in LIST
  - Return to MAINXX

- If RAP = 4:
  - Change resource assignment policy to 2

- If RAP = 5:
  - Change resource assignment policy to 6

- If RAP ≠ 4 or 5:
  - CALL RESAP
  - Was a resource found to serve the search need?
    - Yes: Set SASG; Set RSRC
    - No:
      - Are any other search needs of CASE in QUEUE?
        - Yes:
          - Increase NQUE; Set REA; Set SIGNAL
          - File NOTE in QUEUE
          - Return to MAINXX
        - No:
          - Set TINQ(CASE)
          - SET SFLAG = 1
          - CALL SSS
          - Return to MAINXX
Subroutine SASS

- IDC = 0
- CALL CRES

- IDC = 1
  - J#0
    - yes: IDC = 2
    - no: SNS = time of next sunset

- CALL VEC
  - SNS = time of next sunset

- TOLRS = TOLRS - TIME
  - yes: TOLRS = TOLRS
  - no: TOLRS = TOLRS(PRI(CASE))

- DO, FOR EACH IRS IN RQUE
  - Calculate X
    - Y = END(TYPE(IRS)) - (2.0*X)

- D = END(TYPE(IRS)) + TF(TYPE(IRS))

- STOLR = TOLRS
  - VOP(TYPE(IRS)) = 0
    - yes: STOLR = TOLRS(PRI(CASE))
    - no: SNS = INT(STOLR/D)

- ANS = STOLR - (IP*SNS) - X

- ANS > 0.0
  - yes: ANS = Y
  - no: ANS = 0.0

- ANS = (Y*SNS) + ANS

- ANS < 0.0
  - yes: XMLES < 0.0
    - yes: PR(IRS) = 0.0
    - no: LOOP
  - no: LOOP

- DO, FOR EACH IRS IN RQUE
  - Set COST(IRS) & DVEC(IRS)

- LOOP
SASS Cont.

CSET is empty

no

Remove first IRS from CSET

File IRS in PSET

no

PSET is empty

no

Remove first IRS from PSET

File IRS in RQUE

yes

TSET is empty

no

Remove first IRS from TSET

File IRS in RQUE

yes

RQUE is empty

IDC = 3

yes

Return to calling program

no

Calculate X

File IRS in PSET

no

PR(IRS) < PGR

no

File IRS in CSET

yes

PR(IRS) > PGR

yes

Remove first IRS from RQUE

no

PERC = PDC2

no

SDAY (NOTE) > 1

yes

PERC = PDC1

no

SDAY (NOTE) = 1

yes

PERC = 1.0

no

S2S(CASE) = 2

no

x is the time to vector from the resource's home station to the location of the CASE

(ENDTYP(IRS)) - (2.0 * x) < 0.0

yes

File IRS in TSET

no
Subroutine SSS

IVAR = 0

KLK = 1
no

KLK = 3
no

Is resource type a boat?

DVEC < (SUNSET-TIME)
no

(TIME + DVEC < SUNRISE OF NEXT DAY)
no

File NOTE in LIST

Set RSRC (NOTE = 0)

---

Resource will not be used to serve search need; cancel any upcoming events and CALL EXQ

---

EIAT < 0.0
no

Decrease BCREW by 1

Return to calling program

IB = 0
yes

IB = 1
no

IB = 0, IB = 2

DEP = 0.0
yes

Return to calling program

MFLG = 2
no

Cancel and destroy ONSCN

MFLG = 13
no

Cancel CHEKN Destroy CHEKN

Set MFLG = 9

Call EXQ

Return to calling program

MFLG = 1, 3, 7, 8, 9, 10, 13
MFLG = 11
yes

Print error message

MFLG = 12
no

Cancel READY Destroy READY

MFLG = 9
no

Cancel FUEL Destroy FUEL

MFLG = 5
yes

Cancel SSET Destroy SSET

MFLG = 4
no

Cancel COMPL Destroy COMPL

MFLG = 2
yes

Cancel ONSCN Destroy ONSCN

MFLG = 13
no

MFLG = 1
yes

CALL EXQ

Return to calling program

IB = 0
MFLG = 9
no

RETURN
Endogenous Event COMPL

1. Store COMPL attributes in local variables
2. Destroy COMPL
3. RSRC = 0;
   Decrease SIS by 1
4. IB = 0
   MELG = 9
   IVAR = 0
5. Call EXQ with the idle resource
6. Is the long search completed? no → Return to MAINXX
   yes → Set SIS to -1
7. Are there any non-search needs? no → CALL TERM with the completed CASE
   yes → Is CASE a multi-resource CASE? no → CALL SPAS
       yes → CALL MRAS
8. Return to MAINXX
Endogenous Event SSET

Store SSET attributes in local variables

Destroy SSET

Set RSRC(NOTE) = 0
Set MFLG = 9

SM = SM-(TS*SOA3)

File NOTE in LIST

Set IB=0
IVAR=0

CALL EXQ with the idle resource

Return to MAINXY
Endogenous Event FUEL

- Store FUEL attributes in local variables
- Destroy FUEL
- Update SM(NOTE)
  SM(NOTE) = SM(NOTE) - (TS(NOTE) * SOA3(TYPE(IRS)))
- Calculate TVEC(IRS)
- Set DEP, XDEST and YDEST
- Set HCREW = 2
- Create HOMEF; Cause HOMEF at TIME + TVEC(IRS)
- HCREW value of 2 indicates resource cannot be interrupted while vectoring home to refuel

That is, calculate time to vector to home station of resource

Return to MAINXX
Endogenous Event HOMEF

Store HOMEF attributes in local variables

Destroy HOMEF

Set XR,YR,DEP, XDEST,YDEST

Create SNDLK; Cause SNDLK at TIME+FP(TYPE(IRS))

Return to MAINXX

Update location of resource
Endogenous Event SNDBK

Store SNDBK attributes in local variables

CALL SASS

DO, FOR EACH IR IN ROME

IF B(IR) > N

IF ELAT = 0.0

CALL ROCA

IF SCREW = 0

LOOP

A

File NOTE in LIST

Destroy FLT

Decrease BCREW by 1

B

Set DVEC(IR)

SN = AINT(TIME) + RISE - XRX

TIME > SN

yes

no

SN = AINT(TIME) + SET

TIME + DVEC + AINT(TIME) + RISE

no

no

no

yes

yes

yes

yes

yes

no

no

SN = AINT(TIME) + RISE + 1.0

TIME > SN

no

B

DVEC(IR) < (SN - TIME)

no

SN = AINT(TIME) + RISE + 1.0

TIME + DVEC(IR) > SN

B

ELAT = 0.0

no

no

yes
Endogenous Event SPISE

LIST is empty
no
Remove first NOTE from LIST
I = 0
DO, FOR EACH II in SRISE(ESAC(NOTE))
SASG(II)=1
no
LOOP
I=0
yes
no
I=1
yes
no
DO, FOR EACH II in SRISE(ESAC(NOTE))
Increase SDAY(II) by 1
SASG(II)=0
LOOP
CALL SASS
JRAP=RAP
RAP=4
yes
no
RAP=5
yes
JRAP=2
JRAP=6
IRES=0
yes
no
Increase NQUE by 1; Set REA
Set TINQ; Set SIGNL
File NOTE in QUE
Call RESAP
Call SSS (NOTE,IRES,2)
SFLAG(II)=1
SFLAG(NOTE)=1
II = 0
II = 1
DO, FOR EACH II in SRISE(ESAC(NOTE))
SFLAG(II)=1
II = 0
yes
SFLAG(NOTE)=1
II = 1
no
no
no
Call SSS (NOTE,IRES,1)
Endogenous Event XSET

1. **CQUE is empty**
   - **yes**
   - **Return to MAINXX**
   - **no**
     - **DO, FOR EACH I in CQUE**
       1. **ELG(I) ≠ 3**
          - **yes**
          - **Remove I from CQUE**
          - **Update TQUE(ESAC(I))**
          - **Update TINT(ESAC(I))**
          - **File I in LIST**
          - **LOOP**
          - **no**

2. **Cause XSET at TIME+1.0**
Subroutine SERVE

IRS = INS(CASE)

if IRS = 0

Create FLT
TFLT(IIRS) = 0
TOM(IIRS) = 0
HCREM(IIRS) = 0

ACASE(IIRS) = CASE
FITON(IIRS) = INOTF
ROS = TIME + TVEC + DLYA

if INOTF = 0

RLS = ROS(IIRS) + OST(INOTF)

if INOTF = 0

RLS = ROS(IIRS) + OST(CASE)

no

NO

PRIOR(CASE) = PRI(NOTF)
KRES(NOTF) = IRS

FLAT = 0

no

Create FLI
FLT(IIRS) = 0
TOM(IIRS) = 0
HCREM(IIRS) = 0

ACASE(IIRS) = CASE
FITON(IIRS) = INOTF
ROS = TIME + TVEC + DLY

if INOTF = 0

RLS = ROS(IIRS) + OST(INOTF)

if INOTF = 0

RLS = ROS(IIRS) + OST(CASE)

no

yes

Create DELAY
Cause DELAY at TIME + DLY

IR = INS(CASE)

DEP(IIRS) = 0

if DEP(IIRS) = 0

GO TO (100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, MFLG(IIRS))

MFLG(IIRS) = 11

Return to calling routine

MFLG = 6

Cancel & Destroy FUEL

MFLG = 7

Cancel & Destroy HOME

MFLG = 8

Cancel & Destroy HOMER

MFLG = 9

Cancel & Destroy DELAY

Cancel & Destroy ARSV

XR(IIRS) = XI(IIRS)
YR(IIRS) = YI(IIRS)

MFLG = 1

Cancel ONCN

Destroy ONCN

MFLG = 2
ENDOGENOUS EVENT DELAY

1. Store attributes in local variables
2. Destroy DELAY
3. XDEST(IFLT(I)) = NC
   YDEST(IFLT(I)) = NC
4. IEP(IFLT(I)) = TIME
   TFLT(IFLT(I)) = TIME
5. TEMP = ROS(IFLT(I))
6. TEMP < TIME (yes)
5a. TEMP = TIME
7. Create ARVS
   Cause ARVS at TEMP
8. MFLG(IFLT(I)) = 1
9. Return to MAINXX
ENDOGENOUS EVENT ARVSN

- \( d = \text{SORT} \left( (X - X_{\text{HAND}}) x x 2 + (Y - Y_{\text{HAND}}) x x 2 \right) \)
- \( X_{\text{DEST}} = X_{\text{HAND}} \)
- \( Y_{\text{DEST}} = Y_{\text{HAND}} \)

- \( \text{NEED} (\text{INOTF}) = 17 \)
- \( \text{DEP} = \text{TIME} + \text{THOOK} \)
- \( \text{OST} (\text{CASE}) = 0 / \text{TSP2} + \text{THOOK} \)
- \( \text{TOWSP} (\text{IFLT}) = \text{TSP2} \)

- \( \text{OST} (\text{CASE}) = 0 / \text{TSP1} + \text{THOOK} \)
- \( \text{TOWSP} (\text{IFLT}) = \text{TSP1} \)

- \( \text{JFLAG} = 2 \)

- \( \text{JFLAG} = 3 \)

- Call DTD

- \( \text{NEED} (\text{CASE}) = 17 \)
- \( \text{DEP} = \text{TIME} + \text{THOOK} \)
- \( L \leq 26 \)
- \( \text{OST} (\text{CASE}) = 0 / \text{TSP2} + \text{THOOK} \)
- \( \text{TOWSP} (\text{IFLT}) = \text{TSP2} \)

- \( \text{OST} (\text{CASE}) = 0 / \text{TSP1} + \text{THOOK} \)
- \( \text{TOWSP} (\text{IFLT}) = \text{TSP1} \)

- \( \text{JFLAG} = 5 \)

- \( \text{JFLAG} = 2 \)
SUBROUTINE SRCHF(CASE, INOTF, IRS)

SML = TSM(CASE)
CALL SASS (CASE, SML, IDUM, IND)

IND = 0

IRA = RAP
IF IRA = 4, IRA = 2
IF RAP = 5, IRA = 6
Call RESAP (CASE, IRA, IRS)

JRS = 0

IS2(CASE) = IRS

OST(CASE) = OST + TSM(CASE)/SOA3
RLS(FLFL) = TIME + OST(CASE)

IF MMM/00, DEP(FLFL) = DEP*TSV/2

OST(INOTF) = OST + TSM(CASE)/SOA3
RLS(INOTF) = TIME + OST(INOTF)

If NUMM(INOTF) > 0
OST(INOTF) = OST + TSM(INOTF)/2

RETURN to calling routine

OST(CASE) = OST + TSM(CASE)/SOA3
RLS(FLFL) = TIME + OST(CASE)

OST(INOTF) = OST + TSM(INOTF)/2

OST(INOTF) = OST + TSM(INOTF)/2

RETURN to calling routine

 HttpResponseRedirect
The resource is now covering only

A

no

yes

LOOP

CALL RETN
(CASE, IRS)

Return to MAINXX

ID#2

yes

COQUE is empty?

no

Create CHEKN

Store IRS
in RESNO

Cause CHEKN
at TIME+TCHEK

MFLG(1FLT(IRS)
=13

Return to
MAINXX

DO, FOR EACH
ICGN IN COQUE

Check for any queued
need of the same case
that the covering resource
can serve

No queued need was
found for the resource
to serve; create and cause
a CHEKN for the resource.

yes

FLG#2

no

CALL RES
(CASE, NEED, CONTR, IT)

IT=0 or
CONTR=0

no

yes

DO, FOR EACH
IRRES OF ROQUE

RES=IRS

TVRC(IRS)=0.0

IRES(FITON(1FLT(IRS))=0

CALL SQYUE

Return to
MAINXX

ITOL(CASE)=7

CALL WRECK(ICGN)

CALL WRECK(ICGN)

LOOP

LOOP
SUBROUTINE COVER (CASE, IRES)

IW = 0
XRLS = 0

DO, for each NOTIF in NSET (CASE)
IR = KRES (NOTIF)

IF IR = 0 or IR = IRES

MFLG (IFLT) = 2

ELSIF RLS (IFLT (IR)) ≥ XRLS

LOOP

IF IW = 0

IWAIT (CASE) = 0

Return to calling routine

ELSE

XRLS = RLS (IFLT)
IW = IR

RETURN

END
Endogenous Event CHEKN

Store RESNO in IRS

MFLG(TFLT(IRS)) = 9

Destroy CHEKN

ICK=0

Call EXQ(IRS, *ICK)

EXQ is called to check for any need that the covering resource can serve

ICK=1

The resource was assigned to serve a need in the queue; it is no longer the covering resource.

Create CHEKN

Store IRS in RESMD

Cause CHEKN at TIME=TCHKN

MFLG(TFLT(IRS)) = 13

Return to MAINXX

The resource was not assigned to serve a queued need; another CHEKN is created and caused.

Return to MAINXX
SUBROUTINE RETN(CASE, IRS)

INOTF = FITON(IFLT(IRS))

IF (IRS) = 2

NO

IB(IRS) = 0

IF (INOTF) = 0

COMD(INOTF) = 5

CALL EXQ(IRS)

COUNT(CASE) <= MMM(CASE) + NNN(CASE)

CALL TERM (CASE)

IRS = -IS2(CASE)

S2S(CASE) = -S2S(CASE)

PRI(CASE) = PRI(CASE)

MMN = NNN > 1

DO, for each N in NSET(CASE)

PRI(N) = PRI(CASE)

IF RES(N) = 0

PRIOR(N) = PRIOR(CASE)

LOOP

IS2(CASE) = 0

RETURN TO CALLING ROUTINE
SUBROUTINE TERM

NMBRO(NTH)=
NMBRO(NTH)+NQUE
TSVC(CASE)=
TIME-OCUR(CASE)

If it is a weekday, set I=11; if it is a weekend, set I=12

I = 11

I = 12

Y=TSVC(CASE) * 24.0

CATG1(I)= CATG1(I)+1

Y>1.0

Y>2.0

Y>3.0

Y>4.0

Y>5.0

Y>10.0

CATG2(I)= CATG2(I)+1

CATG3(I)= CATG3(I)+1

CATG4(I)= CATG4(I)+1

CATG5(I)= CATG5(I)+1

CATG6(I)= CATG6(I)+1

CATG7(I)= CATG7(I)+1

KCOUNT=KCOUNT-1

NBRCO = NBRCO+1

Print 'CASE TERMINATED'

A
SUBROUTINE TERM(CASE) Cont.

A

yes

STAPE=0

do, for i= (1)(15)

IBUFF(I)=0

if i<10, FBUFF(I)=0

loop

write two records of case attributes

i=11

no

is SHHS(CASE) empty

yes

i=1

is NSET(CASE) empty

no

remove first n from NSET(CASE)

no

write out IBUFF and FBUFF(2) records on STAPE

return to calling routine

yes

remove FIRST N from SHHS(CASE)

if i<15, IBUFF(I) = RESA(N)

i = i+1

destroy note n

return to calling routine

no

destroy case

return to calling routine

no

IBUFF(I)=NEED(N)

IBUFF(I-1)=RESA(N)

FBUFF(I)=OST(N)

FBUFF(I+1)=DELTA(N)

KRES(N) = 0

if KRES(N) = 0

call EXO(KRES)

yes

destroy notif n

no

return to calling routine
ENDOGENOUS EVENT HOME

- IRS=RESNO
  - EDAT(IRS)=0
  - BCRED(STN)=
  - BCRED(STN)=1
  - DESTROY IRS

TFLT (IFLT) > TLAST
- NO
  - TUTIL(IRS)= TUTIL(IRS) + TIME-TLAST
- YES
  - X=XS(STN)
  - Y=YS(STN)
  - DESTROY FLT
  - IFLT(IRS)=0

Return to MAINXX
A search need is being interrupted; send it back through the system; if no idle resource can serve the need, queue it.

Update SN

Do, for each resource in QUEUE:

- IF IB(IRES) ≠ 0
  - IF EAT(IRES) = 0.0
    - IF HCREM = 0
      - CALL SSS
    - ELSE
      - CALL ROCA
  - IF EAT(IRES) ≠ 0.0
    - RETURN to calling program

File KDT in CQUE

Return to calling program

Set RSRC = IRES

Loop
SUBROUTINE QUEUE (NODE.M)

IF M=1, NOTIFY=FLTON(1FLT(NODE.M))
CASE=CASE(NOTIFY)

NOTIFY=NODE

NEED(NOTIFY) = 17
not

M=0

yes

CALL GRES

no

or

II=0

or

NTR=0

no

CALL VEC

CALL OSET

RETURN TO CALLING ROUTINE

ITOL(CASE)=5

RETURN TO CALLING ROUTINE

IFR(IRS) = 0

yes

CALL ROCA

no

ival=2

yes

CALL SERVE

no

LOOP

RETURN TO CALLING ROUTINE
CALL CRES

DO, FOR EACH IR IN RQUE

IR=IRES

LOOP

Need being considered is a long search need?

Are there any small vessels capable of serving the need that can meet the endurance check?

Yes

Return to calling routine

No

Are there any small vessels capable of serving the need?

Yes

RETURN TO VECTOR TO QUEUED CASE

No

SUBROUTINE EXQ Cont.

A multi-resource need is being removed from COQUE to be served; SVQUE is called

IVAR=0

IWAIT=0

KRES=0

yes

Calculate COST(IRES)

Set PRIOR, ACASE, FITON

CALL SVQUE

Set PRIOR and ACASE

Remove case from COQUE

CALL SERVE

Return to calling routine

A single resource case is being removed from COQUE to be served.

IVAR=0

 FIG=3

yes

Calculate TOQUE & TINT

Update TOQUE & TINT

FIG=2

no

IVAR=0

IWAIT=0

KRES=0

no

Type of resource is a cutter or airplane; however the priority is greater than TPRI

TI=1

Do, FOR EACH IR IN RQUE

SQTAG(TYPE(IRES))

Type of the resource is a small vessel; let it serve the queued need.

no

yes

no

SQTAG(TYPE(IRES))

II=0

Small vessels are not capable of serving the need; let the cutter or airplane serve the need

no

yes

no

yes

Loop

Loop

II=1

FLG=3

FLG=2

Loop

Loop

B

B
A search need is being removed from QUEUE to be served.

\[ \text{XY} = \text{Distance between the location of the need and the location of the station of the resource} \]

\[ \text{End}((\text{TYPE}(\text{IRES})) \leq (2.0 \times \text{XY})) \]

**Yes** \( \rightarrow B \)

**No**

\[ \text{SUNSET < TIME < SUNRISE} \]

**Yes** \( \rightarrow E \)

**No**

\[ \text{TVEC < (SUNSET - TIME)} \]

**Yes** \( \rightarrow E \)

**No**

\[ \text{(TIME - TVEC) > next sunrise} \]

**Yes** \( \rightarrow E \)

**No**

Is resource type an aircraft?

**Yes** \( \rightarrow B \)

**No** \( \rightarrow E \)

Search need will not be served; resource cannot meet endurance check.
Search need will be served

IVAR=0

no

IVAR=0
IWAIT=0
KRES=0

Increase CNRES by 1

Calculate COST and COSTC

Set RSRC

Update TQUE and TINT

SASG=1

Set IB, PRIOR; Increase NCASE by 1

EXQ Continued

Set ACASE, DEP, FITON

Set XDEST, YDEST

Set MFLG=3

Create ARSCH; Cause ARSCH at TIME+TVSEC

Increase NEEDS by 1

Remove search need from CODE

Return to calling program
Endogenous Event STNBY

Store STAT(STNBY) in ISTA

Destroy STNBY

Is CQUE empty? yes

DO, FOR EACH IJK IN CQUE

FLG=1 yes CASE=IJK

FLG=2 yes CASE=CAS(IJK)

FLG=3 yes CASE=ESAC(IJK)

Is this a search need? yes

Is this a multi-resource need? yes

Value of RAP

RAP=1,2,3,6

RAP=4,5

Can station of crew serve CASE?

yes

no

Loop

Increase UNPRO ISTA by 1

Increase TUNPS by 1

A

End

Return to MAINXX

Does ISTA=the primary station or an adjacent to the primary or an aircraft covering station of the primary or a covering cutter of the primary or an aircraft covering station of an adjacent or a covering cutter of an adjacent
SUBROUTINE SVQUE

Case = CAS(NOTIF)
Remove NOTIF from QUEUE
Comp (NOTIF) = 2
KRES(NOTIF) = KRES
N1 = 0 N2 = 0

Do for each NOTIF in NSET (Case)

COMP(NOTIF) = 1

N1 = N1 + 1

SIGNAL(NOTIF) = yes

N2 = N2 + 1

LOOP

yes

N1 = 0

no

yes

N2>0

no

yes

SIGNAL(NOTIF) = 1

no

SIGNAL(NOTIF) = 2

Call Serve

RETURN to calling routine

TME (CASE) = TME + TME - TINO (CASE)

TINT (Case) = TINT + TIME - STINO (CASE)
SUBROUTINE DRIVE

CALL JUMPR
CALL SARSIM
CALL GUBRES
CALL HEADR
DO, FOR EACH STA N

CALL TITLE
DO, FOR EACH J, WITH REST(N,J)=0
CALL RESULT(N,J)
LOOP
LOOP

CALL HEAD
DO, FOR EACH CASE in EXCS
CALL EXCASE
LOOP
CALL DISTRIB

Is COQUE empty ?
yes
Write out sentence saying no CASES in COUE
no
Write out column labels for Queue CASES

DO, FOR EACH IGNN IN COUE
If FLG(IGNN)=1 CASE=IGNN
If FLG(IGNN)=2 CASE=CAS(IGNN)
If FLG(IGNN)=3 CASE=CASE(IGNN)
Write out certain attributes of CASE

Write out footnote to above table

I1=0

DO, FOR EACH RES IRS

I1=1

yes
Write out column labels for busy resources

TB = 0?
no

I1=1?
no
yes

Write out certain attributes of CASE and IRS

LOOP

I1=1

yes
Return to MAINXX
no
Write out sentence saying no busy resources
SUBROUTINE JUMPER

5. WRITE CARD 9

6. LOOP

7. WRITE '5. ATTRIBUTE CHANGE'

8. READ CARD 10

9. CONTENTS CARD 10 EQUAL 0?

10. DO FOR EACH OPEAC WITH ATTRIBUTE CHANGE

11. READ CARD 11

12. WRITE CARD 11

13. LOOP

14. WRITE '"5. attribute change"

15. WRITE ""B RESOURCE TYPE CHANGES"'

16. WRITE ""I. SEE INITIATION DATA FOR..."

17. READ CARD 12

18. CONTENTS CARD 12 EQUAL 0?

19. DO FOR EACH NEW RESOURCE TYPE

20. READ CARD 13

21. WRITE CARD 13

22. LOOP

23. WRITE "NO NEW ADDITIONS"

24. WRITE '2. ATTRIBUTE CHANGES'

25. READ CARD 14

26. CONTENTS CARD 14 EQUAL 0?

27. DO FOR EACH RESOURCE WITH ATTRIBUTE CHANGES

28. READ CARD 15

29. WRITE CARD 15

30. LOOP

31. WRITE "3. CAPABILITY CHANGES"
SUBROUTINE JUMPER

11 READ CARD 16
   CONTENTS CARD 16 EQUAL 0 ?
   no WRITE "SP:
   INITIALIZATION
   DATA FOR...
  yes WRITE "NO
   CAPABILITY
   CHANGES"

12 DO FOR EACH
   RESOURCE TYPE
   WITH CAPABILITY
   CHANGE
   READ CARD 17
   WRITE CARD 17
   LOOP

13 WRITE "C:
   RESOURCE
   INVENTORY
   CHANGES"

14 WRITE "1:
   NEW OPEACS"

15 READ CARD 20
   CONTENTS CARD 20 EQUAL 0 ?
   yes WRITE "NO
   RESOURCE
   CHANGES..."
   no DO FOR EACH
   OPEACS WITH
   INVENTORY
   CHANGE
   READ CARD 21
   WRITE CARD 21
   LOOP

16 WRITE "D:
   CREW MANNER
   LITE CHANGES"

17 WRITE "1:
   NEW OPEACS"
SUBROUTINE JUMPER

17 READ CARD 22
   CONTENTS CARD 22 EQUAL 0 ?
   yes
   WRITE "NO CHANGES"
   no

18 DO FOR EACH NEW OPTAC
   READ CARD 23
   WRITE CARD 23
   LOOP

18 WRITE "2 EXISTING OPTAC"

19 READ CARD 24
   CONTENTS CARD 24 EQUAL 0 ?
   yes
   WRITE "NO CHANGES IN EXISTING"
   no

20 DO FOR EACH OPTAC WITH MANNING LEVEL
   READ CARD 25
   WRITE CARD 25
   LOOP
   WRITE "USER INPUT OPTAC"

21 WRITE CARD 27
   READ CARD 31
   WRITE "2 POSTPROCESSOR"
   WRITE "SPECIAL OUTPUT VIA QUICK QUERY"

21 WRITE CARD 31
   READ CARD 32
   CONTENTS CARD 32 EQUAL 0 ?
   yes
   WRITE "NO SPECIAL REPORT"
   no

22 DO FOR EACH SPECIAL REPORT

23 24