MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0356191 9

ORNL/CSD/TM-269/R1

ornl

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

Scheduling Algorithm for Improving Lift (Sail)

Documentation Revision—August 1991

J. E. Hawthorne
R. A. McLaren

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.
UCN-7969 (3 9-77)

MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/CSD/TM-269/R1

Computing and Telecommunications Division

SCHEDULING ALGORITHM FOR IMPROVING LIFT (SAIL)

DOCUMENTATION REVISION – AUGUST 1991

J. E. Hawthorne and R. A. McLaren

Date Published - August 1991

Prepared for
Military Sealift Command
Interagency Agreement #1521-1521-A1

Prepared by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
managing the

Oak Ridge National Laboratory
Oak Ridge K-25 Site
Oak Ridge Y-12 Plant
Paducah Gaseous Diffusion Plant

for the
DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0356191 9

TABLE OF CONTENTS

Page

LIST OF FIGURES xi

LIST OF TABLES xi

ACKNOWLEDGEMENTS xiii

ABSTRACT xv

EXECUTIVE SUMMARY xvii

1. INTRODUCTION 1

2. MILITARY CRISIS ACTION PLANNING - BACKGROUND 3
   2.1 ORGANIZATIONS AND PLAN DEVELOPMENT 3
   2.2 SAIL DEVELOPMENT 4
   2.3 CONTINUING DEVELOPMENT ACTIVITIES 5

3. THE SEALIFT PROBLEM 7
   3.1 DELIBERATE AND EXECUTION PLANNING 7
   3.2 DETERMINISTIC MODELS AND THEIR IMPACT ON ASSESSMENT 8
   3.3 INTERFACES OF SEALIFT PLANNING WITH OTHER MEMBERS OF THE JOINT DEPLOYMENT COMMUNITY 9
   3.4 INPUTS TO THE SEALIFT PLANNING SYSTEM 9
      3.4.1 Cargo Data 10
      3.4.2 Ship Data 10
      3.4.3 Port and Other Facility Data 10
      3.4.4 Distance Computation Data 11
      3.4.5 Timing Information 11
      3.4.6 Other Data 11
   3.5 THE RESULTS OF THE SEALIFT PLANNING MODEL 11
      3.5.1 The Assets Required 12
      3.5.2 The Deployment Schedule 12
      3.5.3 Analysis of Alternatives 12

4. THE SEALIFT SCHEDULING MODEL 13
   4.1 THE SCHEDULING CONCEPT USED IN SAIL 13
   4.2 AN OUTLINE OF THE SOLUTION PROCEDURE 14
      4.2.1 The Database Management System Interface 14
      4.2.2 Cargo Aggregation 15
      4.2.3 Distance Computations 15
      4.2.4 Ship Availability Dates 18

Page

4.2.5 The Ship Routing Subsystem 18
4.2.5.1 Optimization-Based Routing 19
4.2.5.2 The Heuristic Routing Approach 21
4.2.5.3 Ship Routing Summary 22
4.2.6 The Ship Loading Subsystem 22
4.2.7 Simulation Subsystem 23
4.2.8 The Reporting Subsystem 24
4.3 SPECIAL SCHEDULING CONCERNS AND PROBLEM FEATURES ... 25
4.3.1 Fleet Use Restrictions 25
4.3.2 Ship Size Restrictions 25
4.3.3 Container Ship Facility Requirements 25
4.3.4 Variable Draft (Future) 25
4.3.5 Queuing at Ports 26
4.3.6 Port Availability Dates 26
4.3.7 Port Facility Throughput Changes 26
4.3.8 Preferred Ship Loading 26
4.3.9 Flexible Cargo Loading 26
4.3.10 Unit Integrity (Mainly Future) 26
4.3.11 Ammunition Routing Restrictions 27
4.3.12 Changes in Canal States 27
4.3.13 Delays at Canals 27
4.3.14 Intermediate Stops 27
4.3.15 Attrition 28
4.3.16 Ocean Avoidance 29
4.3.17 Underutilized Ships 29
4.3.18 Assault Follow-On Echelon (AFOE) 29
4.3.19 At-Sea Unloading of Resupply Petroleum, or Consolidation of Load (CONSOL) 30
4.3.20 Bunkering 30
4.3.21 Unacceptably Late Cargo 30
4.3.22 The Refine Module 30
4.3.23 Minimum Port Activity 31
4.3.24 Special Treatment of Non-Combatant Evacuees (NEO) 31
4.3.25 Lift Capability Computations 31
4.3.26 Required Lift Analysis 31
4.3.27 Multiple Plan Capability 32
4.3.28 Special Scheduler for Sparse Shipping 32
4.3.29 The Transfer Port Concept 32
4.3.30 Experimental Scheduling from CONUS Complexes 33
4.4 SOME COMMENTS ON COST FUNCTIONS 33

5. THE SAIL MODEL - IMPLEMENTATION 35
5.1 THE SAIL SOFTWARE 35
5.1.1 The MAIN Program 35
5.1.1.1 Subroutine INIDIM 36
5.1.1.2 Subroutine ROUTCON 36

Page

5.1.1.3 Subroutine REDSOL 36
5.1.1.4 Subroutine WRTSOL 36
5.1.1.5 Logical Function INFEAS 36
5.1.1.6 Subroutine SOLU 37
5.1.1.7 Subroutine DELETE 37
5.1.1.8 Subroutine NITIAL 37
5.1.1.9 Subroutine INTHRU 37
5.1.1.10 Integer Function ANYCAP 37
5.1.1.11 Subroutine PORCHK 37
5.1.2 The Interface Subsystem 37
5.1.3 The Aggregation Subsystem Routines 37
5.1.3.1 Subroutine SPLIT 37
5.1.3.2 Subroutine ELIM 38
5.1.3.3 Subroutine INTERV 39
5.1.3.4 Subroutine PODQ 39
5.1.3.5 Subroutine POEQ 39
5.1.3.6 Subroutine CHANEL 39
5.1.3.7 Subroutine WREXCL 40
5.1.3.8 Subroutine FINDIN 40
5.1.3.9 Subroutine INITCH 40
5.1.3.10 Subroutine CLUSTR 40
5.1.3.11 Subroutine PCLUS 40
5.1.3.12 Subroutine CULL 40
5.1.4 The Distance Computation Subsystem 41
5.1.4.1 Subroutine DISCAL 41
5.1.4.2 Subroutine SADS 41
5.1.4.3 Subroutine DISCHK 41
5.1.4.4 Subroutine CANFND 41
5.1.4.5 Integer Function DIST 41
5.1.4.6 Subroutine SPA 41
5.1.4.7 Subroutine INLINK 41
5.1.4.8 Integer Function GCD 42
5.1.4.9 Subroutine NEAR 42
5.1.4.10 Subroutine STAR 42
5.1.4.11 Subroutine SAVDIS 42
5.1.4.12 Subroutine GETDIS 42
5.1.4.13 Subroutine ADPORT 42
5.1.4.14 Subroutine INNODE 42
5.1.4.15 Subroutine DISFIL 43
5.1.4.16 Logical Function NTRSEX 43
5.1.4.17 Subroutine INLINE 43
5.1.4.18 Subroutine DISTIM 43
5.1.4.19 Logical Function COPEN 43
5.1.4.20 Subroutine TABLED 43
5.1.4.21 Subroutine CROUTE 43

Page

5.1.4.22 Subroutine SAD1 43
5.1.4.23 Subroutine RISKS 44
5.1.4.24 Subroutine DECOMP 44
5.1.4.25 Subroutine HYPERB 44
5.1.4.26 Subroutine INIRSK 44
5.1.4.27 Subroutine LNKRSK 45
5.1.4.28 Subroutine RISKIT 45
5.1.4.29 Subroutine PTOP 45
5.1.4.30 Subroutine TOPOES 45
5.1.4.31 Subroutine CASE1 45
5.1.4.32 Subroutine SADTAB 45
5.1.4.33 Subroutine INIDIS 45
5.1.4.34 Subroutine ARC 45
5.1.4.35 Subroutine AVOID 45
5.1.4.36 Subroutine INVIEW 46
5.1.4.37 Real Function UNTIME 46
5.1.4.38 Subroutine FNTIME 46
5.1.4.39 Integer Function DISBAS 46
5.1.4.40 Real Function RISKY 46
5.1.4.41 Real Function RISTIM 46
5.1.4.42 Subroutine DISCPY 46
5.1.4.43 Subroutine SNEAR 47
5.1.4.44 Subroutine COMNOD 47
5.1.4.45 Subroutine SADDUN 47
5.1.4.46 Subroutine KEEPRT 47
5.1.4.47 Subroutine SHPRTE 47
5.1.4.48 Subroutine INWORL 47
5.1.5 The Ship Routing Subsystem 47
5.1.5.1 Subroutine ACYCLE 47
5.1.5.2 Subroutine APRIME 48
5.1.5.3 Subroutine ASAILM 48
5.1.5.4 Subroutine NCYCLE 50
5.1.5.5 Subroutine NPRIME 50
5.1.5.6 Subroutine NSAILM 50
5.1.5.7 Logical Function FULL 50
5.1.5.8 Subroutine STASH 50
5.1.5.9 Subroutine NXTBST 50
5.1.5.10 Subroutine EVLATE 50
5.1.5.11 Subroutine TEMPLD 51
5.1.5.12 Subroutine PURGE 51
5.1.5.13 Subroutine COLSOL 51
5.1.5.14 Subroutine MATGEN 51
5.1.5.15 Subroutine MATREV 52
5.1.5.16 Subroutine TPCOST 52
5.1.5.17 Subroutine COLPTR 53
5.1.5.18 Subroutine FIRSTX 53

Page

5.1.5.19 Subroutine SOLVE 53
5.1.5.20 Subroutine SORTOF 53
5.1.5.21 Subroutine RCGEN2 54
5.1.5.22 Subroutine SORTRC 54
5.1.5.23 Subroutine RCROW 54
5.1.5.24 Subroutine RCCOL 54
5.1.5.25 Subroutine DEPCHK 54
5.1.5.26 Subroutine SECON1 54
5.1.5.27 Subroutine CHINIT 54
5.1.5.28 Subroutine CHEVAL 55
5.1.5.29 Subroutine PORD 55
5.1.5.30 Function LODTIM 55
5.1.5.31 Function UNLTIM 55
5.1.5.32 Subroutine WOODEN 55
5.1.5.33 Subroutine DONE 55
5.1.5.34 Subroutine QWAIT 55
5.1.5.35 Subroutine COLSET 56
5.1.5.36 Subroutine ROWGET 56
5.1.5.37 Subroutine ADDCHN 56
5.1.5.38 Subroutine BLDSOL 56
5.1.5.39 Subroutine ACCEPT 56
5.1.5.40 Subroutine REJECT 56
5.1.5.41 Subroutine OPTION 56
5.1.5.42 Subroutine TEST 56
5.1.5.43 Subroutine OKNEO 57
5.1.5.44 Function URAND 57
5.1.5.45 Real Function RISK 57
5.1.5.46 Subroutine SHPSEL 57
5.1.5.47 Subroutine MISMAT 57
5.1.5.48 Subroutine SOLU 57
5.1.6 Ship Loading Subsystem 57
5.1.6.1 Subroutine ARRDEP 58
5.1.6.2 Subroutine GNCDAT 58
5.1.6.3 Subroutine GNC 58
5.1.6.4 Subroutine MGNC 58
5.1.6.5 Subroutines GFxxx 58
5.1.6.6 Subroutine ENRICH 60
5.1.6.7 Real Function GTCOST 60
5.1.6.8 Subroutine SHRANK 61
5.1.6.9 Subroutine GNCOUT 61
5.1.6.10 The General Network Solution Routines 61
5.1.7 The Simulation Subsystem 61
5.1.7.1 Subroutine SIMOVR 64
5.1.7.2 Subroutine SIMULA 64
5.1.7.3 Subroutine SKEDIN 64
5.1.7.4 Subroutine INXTRA 65

Page

5.1.7.5 Subroutine ASSURE 65
5.1.7.6 Subroutine PATH 65
5.1.7.7 Subroutine CENTER 65
5.1.7.8 Subroutine FAREST 66
5.1.7.9 Subroutine CHECK 66
5.1.7.10 Subroutine REPATH 66
5.1.7.11 Subroutine STOPS 66
5.1.7.12 Subroutines SWPR and EXCH 66
5.1.7.13 Subroutine PATHD 66
5.1.7.14 Subroutine CHKCH 66
5.1.7.15 Subroutine CHKAMO 66
5.1.7.16 Subroutine EMBARK 66
5.1.7.17 Subroutine DEBARK 67
5.1.7.18 Subroutine STATUS 67
5.1.7.19 Subroutine UNDEL 67
5.1.7.20 Subroutine PORTTM 67
5.1.7.21 Subroutine CHKCAP 68
5.1.7.22 Subroutine INSERT 68
5.1.7.23 Subroutine ADEQAT 68
5.1.7.24 Subroutine SAILIN 68
5.1.7.25 Subroutine FILLCP 68
5.1.7.26 Subroutine PORCHK 68
5.1.7.27 Subroutine SRTULL 68
5.1.7.28 Subroutine CHANGE 68
5.1.7.29 Subroutine NOSTOP 69
5.1.7.30 Subroutine ENUF 69
5.1.7.31 Subroutine REMOVE 69
5.1.7.32 Subroutine NXCSQF 69
5.1.7.33 Subroutine NXCP 69
5.1.7.34 Subroutine NXCMT 69
5.1.7.35 Subroutine OSCOMP 69
5.1.7.36 Subroutine FINDCP 69
5.1.7.37 Subroutine SLPCHK 69
5.1.7.38 Subroutine STATUS 70
5.1.7.39 Subroutine ALDCHK 70
5.1.7.40 Subroutine ENDSIM 70
5.1.8 The Refinement Process 70
5.1.8.1 Subroutines REFIN1 and REFIN2 70
5.1.8.2 Subroutine DUMSHP 70
5.1.8.3 Integer Function LV 71
5.1.8.4 Subroutine INIMRK 71
5.1.8.5 Subroutine COMPAT 71
5.1.8.6 Subroutine NEWROW 71
5.1.8.7 Subroutine FRACLD 71

Page

5.1.9 The Bunker Routines 71
5.1.9.1 Subroutine FUELS 71
5.1.9.2 Subroutine PITSTP 71
5.1.9.3 Subroutine FILL 71
5.1.10 The Reporting Subsystem 72
5.1.10.1 Subroutine POST 72
5.1.10.2 Subroutine REPTC 72
5.1.10.3 Subroutine REPTS 72
5.1.10.4 Subroutine REPTP 72
5.1.10.5 Subroutine REPTSC 72
5.1.10.6 Subroutine STATS 73
5.1.10.7 Subroutine CARSTO 73
5.1.10.8 Subroutine HISTO 73
5.1.10.9 Subroutine PWRREP 73
5.1.10.10 Subroutine LCREP 73
5.1.10.11 Subroutine RLREPT 73
5.1.10.12 Subroutine RISKCS 73
5.1.11 Subroutines to Support the Transfer Port Concept 73
5.1.11.1 Subroutine SETPOX 74
5.1.11.2 Subroutine INPOX 74
5.1.11.3 Subroutine POXSAV 74
5.1.11.4 Subroutine FORBDN 74
5.1.11.5 Subroutine POXLAD 74
5.1.11.6 Subroutine RESTOR 75
5.1.11.7 Subroutine MAKEPOI 75
5.1.11.8 Subroutine SETSAD 75
5.1.12 Lift Capability Analysis 75
5.1.12.1 Subroutine LCINPT 75
5.1.12.2 Subroutine LCYCLE 75
5.1.12.3 Subroutine LPRIME 75
5.1.12.4 Subroutine LSAILM 75
5.1.12.5 Subroutine LSTASH 76
5.2 INPUT AND OUTPUT FILE STRUCTURES 76
5.2.1 Blocks of Code and the Subsystems 76
5.2.2 File Usage Table 76
5.2.3 Description of Input Files 78
5.2.3.1 Major Data Files 78
5.2.3.2 Parameter Files 83
5.2.3.3 Standard Reference Data Files 89
5.2.4 Descriptions of Output Files 94
5.2.4.1 Internal System and Development Files 94
5.2.4.2 Major System Interface Files 96
5.2.4.3 Files for Plotting Routes 98
5.3 PARAMETERS AND OPTIONS 99
5.3.1 Time Division and Aggregation Parameters 99
5.3.2 Cost Coefficients (or Penalties) 100

Page

5.3.3 Ship Routing Parameters 102
5.3.4 Ship Operating Parameters--Initial Estimates 104
5.3.5 Model Features Control Parameters 105
5.3.5.1 Canal Restrictions 105
5.3.5.2 Port Dimensional Constraints 106
5.3.5.3 Gross Control of the Available to Load Date (ALD) 106
5.3.5.4 Delays at Intermediate Ports 106
5.3.5.5 Pseudo-Random Number Generation Seed 107
5.3.5.6 Control of Underutilized Shipping Resources 107
5.3.5.7 Route Refinement 107
5.3.5.8 Bunkering 107
5.3.5.9 Restricted Use of the Korean and NATO Fleets 108
5.3.5.10 Control of Optimization Procedure 108
5.3.5.11 Control of Solution Strategy 108
5.3.6 File Names to Save Intermediate Model Results 109
5.4 PROGRAM DIMENSIONS 110
5.5 PROGRAMMING SUBSCRIPT CONVENTIONS 112
5.6 OTHER CODES AND CONVENTIONS 115

ACRONYMS AND TERMS 119

REFERENCES 123

APPENDICES 125

LIST OF FIGURES

Figure	Page
4.1. Part of an ocean network	16
5.1. Acceptance criteria for secondary channels	49
5.2. The ship-loading problem - a generalized network formulation	59
5.3. Illustration of cost function for the ship-loading problem	62

LIST OF TABLES

Table	Page
5.1. Input Program Descriptions	38
5.2. General Network Subroutines	63
5.3. SAIL File Map	77
5.4. Array Dimension Sizes	111
5.5. Ship Type to Integer Map	113
5.6. Cargo Class to Integer Map	113
5.7. Port Facility Type to Integer Map	114
5.8. Ship Stowage Type to Integer Map	114
5.9. Fleet Codes	115
5.10. Exclusion Codes	116
5.11. Routing Cost Function	117
5.12. General Network Cost Function	118

ACKNOWLEDGEMENTS

Previous editions of this document have recognized the contributions to the SAIL algorithm made by many people in several organizations. The staff of the Military Sealift Command have been of particular importance throughout the development. But in this revision, we shall not repeat our prior indebtedness in detail; instead, we recognize some new names of those recently influential in the work.

Locally, we wish to thank Messrs. Brian Jones and Robert Edwards who have integrated the SAIL algorithm into U.S. Transportation Command's *FAST* system. The exposure of the algorithm to a different operating environment has revealed new opportunities for exploiting SAIL. For example, the use of SAIL to provide required lift estimates is an outgrowth of this interaction. We are also newly indebted to Ms. Debbie Bain for her assistance in the preparation and publication of this edition.

We also thank the staff of the U.S. Transportation Command who, through the *FAST* system, employed SAIL in analyzing various problems during Desert Shield/Desert Storm. This process provided an important element of realism in assessing the utility of SAIL which may not be duplicated for years. A special thank you goes to Lt. Col. Robert D. Hunter; his energy and insight were of particular importance in this shake-down process.

SCHEDULING ALGORITHM FOR IMPROVING LIFT (SAIL)

DOCUMENTATION FOR INITIAL OPERATING CAPABILITY

J. E. Hawthorne and R. A. McLaren

ABSTRACT

The Military Sealift Command, a component of the United States Transportation Command, is responsible for the sealift of military personnel and materiel during a crisis. Conceptual plans for these complex moves, called deliberate plans, are continually being prepared. A computer-based scheduling system, the Sealift Strategic Analysis Subsystem (SEASTRAT), is under development for assisting in the production of these plans. The ship scheduling portion of this system, the Scheduling Algorithm for Improving Lift (SAIL), combines linear optimization and heuristic methods to determine ship routes and cargo loadings which honor a variety of complex operational constraints.

EXECUTIVE SUMMARY

The United States Transportation Command is made up of three component commands which share the responsibility of moving military personnel and materiel during a crisis. The Military Sealift Command manages the common-user ocean shipping portion of this activity. Since crisis-related military movements are rare, an integral part of this responsibility is the establishment of an infrastructure in which to carry out the movements, should they be required. Major planning activities are conducted continually to reassess the resources needed to support a wide variety of potential military options.

This advance analysis is called deliberate planning. The process involves a consideration of the massive amounts of materiel that must suddenly be moved, the variety of resources available to move it, and the intricate timing involved in a successful military operation. The volume of data involved in this analysis suggests a need for automated aids to the planning process. The Sealift Strategic Analysis Subsystem (SEASTRAT) is the Navy's new computing system designed to support the deliberate planning process; this report deals with SEASTRAT's ship scheduling algorithm.

The objective of the scheduler is to find an allocation of shipping resources that delivers the sealift portion of the movements on time and with effective utilization of resources. Naturally, it is not always possible to meet this broad objective; there are many subobjectives in problems of this type, some of which may conflict with one another or with the primary objectives. The design challenge of this system derives from these conflicts. Both the concept and solution methods are impacted by the compromises required to resolve the conflicts, as will be evident throughout this report.

The scheduler is divided into several modules. The early modules accept data which has been gathered, arranged, and edited by the SEASTRAT data management system; interport distances are computed by using a world network; ships are located on this network and their availability dates at ports are computed; and cargos are aggregated to reduce the size of the problem.

The next module is the key to the system: it develops routes for ships. To begin with, there are an extraordinary number of possible routes for the ships; some procedure must be used to pare down this number to a relatively few routes that can be refined. A linear optimization problem is formulated to aid in this process. Such networks provide a way of efficiently dealing with the large number of combinations of possible events in the ship routing problem, but they do not deal with all the constraints which define how cargo can be combined and shipped. Decision rules are used to represent these more complicated constraints. The scheduler uses these two techniques alternately until all ships routes are established.

This optimization-based routing system permits simultaneous evaluation of many options and thus gives some intuitive assurance that short-sighted allocations of resources have been avoided. The penalty paid for this assurance is computation time. Results to date

show that large planning problems may not be solved in an acceptable time frame, particularly during time-sensitive planning. As a result, an optional heuristic-based routing system has been developed. Limited tests indicate this approach will give results almost as good as the optimal-based method, but in a fraction of the time. The optimal system remains superior for small, resource-constrained problems and it continues to be the standard to which the heuristic process is compared for all problems.

As the routes are developed by either of the routing methods, an allocation of cargos to ships is made. There is an opportunity, however, to better load the ships once the routes are established. A linear programming problem is formulated to accomplish this loading. Objectives are to minimize lateness, use few ships, and load compartments with compatible cargos; the established routes constrain the cargos that can be placed on each ship. All cargos, including those unscheduled in the routing process, have a new opportunity for effective assignments to ships.

When the linear program problem is completed the ships are routed and aggregations of cargos are assigned to the ships. Timing information, disaggregation back to original cargos, and other details are added in a simulation module. These results are then handed back to the data management subsystem of SEASTRAT for the preparation of reports.

1. INTRODUCTION

During the execution of an operation plan, the Military Sealift Command (MSC) is responsible for the common user ocean transportation required to support the needs established by the commander-in-chief of the operation. These movements can be both large and complex. The Command must develop plans for potential emergencies, identify the resources needed to execute an operation, establish the feasibility of the movements, and, in the case of an actual emergency, determine the utilization of shipping resources.

To aid in developing these transportation plans, a new generation planning system is being implemented. The Sealift Strategic Analysis Subsystem (SEASTRAT) is to be a comprehensive data collection, data management, ship scheduling, and reporting tool. This document addresses the ship scheduling portion of that system.

There are several classifications of potential readers: planning personnel, developers of other SEASTRAT systems, software maintenance personnel, and individuals interested in determining how this system fits in with other planning systems throughout the deployment planning community. The several major sections of the report are designed for these various audiences.

Section 2 provides a brief overview of the military crisis planning problem; its purpose is to set the stage for the remainder of the report. In Section 3, the sealift problem is discussed specifically, and some philosophy about planning models in general is included. It goes on to present the current model in terms of its inputs and outputs. In Section 4, the general methodology used for solving the sealift planning problem is described, and for the most part, Section 4 is intended for a general audience. A few paragraphs require a background in optimization methods, which some readers may wish to skip. In Section 5, the software that implements the methods previously presented is described. The section gives information about the overall structure of the computer codes, a summary of how each subroutine works, how to size the code for different problems, and other technical tidbits primarily of interest to software maintenance people. Appendices then fill in some of the details of the methodology; a dictionary of internally used variables is also included.

As this report is written, the SEASTRAT system, including the sealift scheduling system, is being tested and readied for initial operation by the Military Sealift Command. Undoubtedly, changes to the scheduler will be required as a result of these exercises. But even if no alterations were indicated, other activities in the planning community assure that planning support tools will not remain static:

- Work has begun on a crisis management system that is designed to continue the planning process during execution. This means the planning algorithm will need to respond to feedback from actual operations as they occur.

  There is a close connection between the continental United States land movements managed by the Military Traffic Management Command and the sealift movements

controlled by MSC. The task of developing an integrated surface movement system remains to be done.

The United States Transportation Command is interested in assuring that the planning activities of all Component Commands are consistent and increasingly integrated with the total transportation community.

During a crisis, planning is a process that involves not only the Headquarters level, but continues down through Area Commands, and eventually to individual ports. Tools to support this dissemination of the planning function are likely to be required.

The capability of computing hardware and software continues to advance at an astonishing pace. Desk-top computers now have more capability than the mainframes of a couple of decades ago, and the improvements in performance are expected to accelerate. These advances will greatly impact the way transportation planning can be done.

For these and other reasons, planning tools and their documentation are never finished. Constant change will be the only constant in this endeavor. But if there are to be systems, they must be built at the risk of obsolescence, and similarly, reports must be written even if their subject changes before they can be published.

2. MILITARY CRISIS ACTION PLANNING - BACKGROUND

The Department of Defense must be able to react appropriately to a wide variety of crisis conditions worldwide, ranging from small peacekeeping actions to major defense encounters. The amount of coordination required to successfully accomplish these actions, even the small ones, is an extraordinary process of data accumulation, information generation, analysis, and dissemination of decisions. The outline of the crisis action planning process in this section is intended to provide the reader with some background for understanding the role that sealift scheduling plays in this overall system.

2.1 ORGANIZATIONS AND PLAN DEVELOPMENT

Crisis planning generally begins with a directive from the National Command Authority to a military commander to formulate a proposed course of action that meets specified objectives. This commander, known as the supported commander, has primary responsibility for all aspects of a task assigned, including determining the types of military units required, when and where they are needed, and the supplies and support necessary to sustain the operation for some period of time. (See, for example, the *Joint Staff Officer's Guide, 1986.*)

Two other types of military organizations then become involved: those which will supply the personnel and materiel, and those which will provide transportation to the crisis area. This report is concerned with the latter category.

There are three service-specific transportation commands, called Transportation Component Commands (TCCs); during a crisis, or for purposes of crisis planning, these commands report to the recently-formed United States Transportation Command (USTRANSCOM). One component is the Military Traffic Management Command (MTMC), a major command of the U.S. Army, whose responsibilities include the surface transportation of materiel inside the continental United States and the operation of certain common-user port facilities worldwide. The other components are the Navy Military Sealift Command (MSC), which is concerned with ocean shipping, and the Air Force Military Airlift Command (MAC), which has airlift responsibility for the Department of Defense.

These transportation commands contribute to the crisis planning activity by determining the feasibility of moving the required personnel and hardware specified by the supported commander within the time constraints imposed and with the resources allocated. They further determine the actual lift resources required and formulate a plan for all movements.

This planning process has been through several generations of development. In recent years, each transportation command has had a computer model that was designed to partially automate the planning. The individual plans were brought together and eventually

refined using a data system known as the Joint Deployment System, operated by the Joint Deployment Agency or its successor, USTRANSCOM.

All of the planning systems are currently in the process of being upgraded. The previous models, which were rule-based algorithms, have become obsolete along with the computers, support software, data management, and communications networks through which they operate. The Military Sealift Command began a study in the early 1980s which determined the specifications for a new, optimization-based planning model to be supported by a database management system and operated on a dedicated supercomputer. The development of this system was delayed by a potential reorganization of the command, but was revived in 1985.

This new system was initially called SEASTRAT, which stands for Sealift Strategic Analysis Subsystem; later, this work was incorporated in a larger effort called Mobility Planning and Execution (MOPX). The new system includes the required communications to access data, a data management system to organize this information, a sealift planning module, a fuels-analysis module, and a reporting function. The planning module, the Scheduling Algorithm for Improving Lift (SAIL), is the specific subject of this report.

The Navy Regional Data Automation Center (NARDAC), Washington, was given the responsibility for developing SEASTRAT; by interagency agreement with the Department of Energy, the task of developing the scheduling algorithm was then transferred to the Oak Ridge National Laboratory.

2.2 SAIL DEVELOPMENT

Work began by reviewing ongoing research in military transportation planning, particularly focusing on programs at the Massachusetts Institute of Technology and at the Georgia Institute of Technology. From this review, a prototype system was designed to demonstrate that an optimization-based scheme could be employed on the sealift planning problem. The prototype was delivered to NARDAC in the autumn of 1986; results were sufficiently encouraging to justify continued development.

The next increment of work, called Phase I, had the objective of converting the prototype into a product that would be useful to MSC planners. This upgrading involved inserting several key constraints and features that had previously been ignored or handled superficially. The work was documented in the report ORNL/CSD/TM-258, *Scheduling Algorithm for Improving Lift (SAIL) - Phase I Documentation*, dated July 1988.

Post-Phase I effort has primarily involved the migration of the software to the IBM 3090 computer system at NARDAC and the improvement of the software's performance, in terms of both efficiency and plan quality. As anticipated, some adjustment of the code was necessary to properly exploit the vector processing capability of the 3090. Additionally, it was found that other changes were required to get the best use from the high-speed cache memory.

Even after these changes it was found that the ship routing module was unacceptably time consuming. Further steps were taken to improve the efficiency of the solution process. Although these measures resulted in a considerable time savings, the ship routing process might still require too much time on large problems. Therefore, a heuristic routing algorithm was developed as an option to the optimization-based method of Phase I. A description of these two approaches will appear in Section 4.2.5.

Software testing using unclassified data sets revealed the need for certain other changes in the model, such as further constraining the port stops a ship will make. These refinements appear as new features in Section 4.

2.3 CONTINUING DEVELOPMENT ACTIVITIES

As long as complex models, such as SAIL, are in use, they are in revision. Additional capabilities, changes in emphasis, and operational experience all dictate modification. Activities such as Desert Shield/Desert Storm are particularly influential in suggesting change. Presently, there are several thrusts.

Integration of the scheduling algorithm into the day-to-day activities within the transportation organizations for both planning and operations is a top priority. Completing and perfecting user interfaces is the most critical task in this effort. Only after continuing use of the scheduler will deficiencies be identified and improvements suggested.

The interface between sealift scheduling and other transportation activities remains a major challenge which holds the promise of improved use of scarce sealift assets. The emergence of support software being developed for the U.S. Transportation Command has given renewed emphasis to this area of development.

Desert Storm experiences show the need for variations on the scheduling theme. One of these is to be able to find the amount of lift required to support a particular deployment description. The converse to this is to find the amount of materiel that can be moved with a given set of lift resources, given general guidelines such as the routes involved and the mix of cargo classes.

Other needs in Desert Storm were for mechanisms for bringing ships into the plan, the use of which were restricted in various ways. Some ships, for example, could not be used in too close to the theater of operations; others were available for only one trip, or for a particular time span. The need to track ships and forecast arrivals also arose. (This problem can be addressed by making special use of the distance computation features of SAIL.)

Clearly, a continuing effort will be required to incorporate these capabilities into the sealift planning system; indeed, a vital system will be in constant change.

3. THE SEALIFT PROBLEM

The sealift planning problem is a key component of the military's crisis action planning function. In this section, the way in which the SAIL model will contribute to the planning activity described in Section 2, will be outlined. Primary data sources and the products of the model will be described. Finally, the way in which the SAIL model and the SEASTRAT system interface with other parts of the military transportation planning system will be briefly indicated.

The discussion is not meant to be a thorough description of these complicated planning functions; rather, it should provide the reader with enough background about the sealift function to understand the motivation behind the structure and features of the SAIL model. We begin in Paragraph 3.1 with definitions of two important types of military planning and continue in 3.2 with a discussion of the strengths and weaknesses of two kinds of modeling approaches.

3.1 DELIBERATE AND EXECUTION PLANNING

Deliberate Planning is the term used to describe the activities concerned with the detailed planning for potential military involvement in crisis situations throughout the world. The purposes of such planning include identifying forces and resources needed to react to a particular crisis, discovering how the many military and civilian organizations involved would need to interact, and gaining insight into the feasibility of the plan. Clearly, the ability to move the personnel and materiel to the crisis area in a timely manner is a key element in this assessment. For the sealift problem, Deliberate Planning is the detailed examination of the movements required by ocean carriers, including the number of ships required, ports to be used, pickup and delivery times, and the effective utilization of shipping resources.

During Execution Planning an approved operation plan or other course of action is adjusted and refined to fit a prevailing situation and is converted into an executable Operation Order. Execution Planning differs from Deliberate Planning in three important ways; first, the crisis is assumed to be much more imminent, resulting in a decreased time available for planning by the participants and the coordination of those plans among the organizations involved. Second, specific details may replace notional information; also, the implementation of the plan may be proceeding while planning continues, resulting in real-world data being fed back into the planning process. And third, the process is focused on immediate operations -- determining what needs to be done in the field today.

Additional introductory information concerning these planning modes can be found in JCS Pub. 15 and AFSC Pub. 1, which are listed in the references.

The SAIL model was originally conceived to be a Deliberate Planning tool. During the long period between the concept's origin and the current implementation, however, emphasis within the planning community has been shifting toward Execution Planning. Until

recently, the latter concept had not been adequately defined to permit model development. Thus, SAIL was constructed around its original specifications--Deliberate Planning--but with an eye toward the modifications necessary to convert to an Execution Planning model in the future.

Consideration of the evolution to Execution Planning is now under way. There are at least two basic notions about how such planning should proceed. One looks at the planning activity as a continuous process; as new data arrive, they are factored into an on-going planning computation. The other approach is to re-plan at discrete points in time, keeping intact some portions of the results from the preceding planning cycle. In either case, the relevant objective is to allocate the uncommitted shipping resources to the unsatisfied movement needs.

If SAIL is used in Execution Planning, it will operate in the latter mode. The principal modification required will involve partially committed resources, such as ships that are partly filled at the time rescheduling occurs. The status of the ship must be represented to the scheduler in such a way that its on-going activity is not interrupted; only its future course may be subject to alteration.

3.2 DETERMINISTIC MODELS AND THEIR IMPACT ON ASSESSMENT

Planning models can be divided into two major types: deterministic and stochastic. Deterministic models resemble the way an individual may plan for a day's activities, setting up a prescribed amount of time for each event. A doctor's appointment calendar is such a plan. These kinds of schedules are clearly essential elements in conducting all types of business and social activities.

However, deterministic plans omit a key ingredient of reality: randomness. It is randomness that prevents the doctor's appointment calendar from being an accurate predictor of when a patient will actually see the doctor. And randomness will, more often than not, cause delays in the execution of a deterministic plan.

Stochastic models are those models that recognize and deal with randomness in the execution of plans. They are generally simulation models, in which the events of the plan take on randomly generated attributes. For example, the time to travel a particular distance by sea is a function of weather, the ship's condition, potential enemy action, etc. Since these conditions cannot be forecast with accuracy, the ship's travel time may be expressed as a probability distribution function; it will take, on the average, a given number of hours to make the trip, but there are assumed odds that it will take more or less time.

The results of a stochastic planning model differ from a deterministic plan in two important respects. First, the stochastic plan will not say, as a deterministic model would, that a ship will arrive on a particular date; it will say that the ship has various chances of arriving on each of several days. Secondly, and more importantly, the natural optimism of deterministic planning is avoided; the impact (usually detrimental) of randomness is accounted for in the results.

SAIL is a deterministic planning tool. It is thus suited for the same kinds of uses as the doctor's appointment calendar: it sets up a list of things that must happen, it prescribes target times for those things to happen, and it determines the resources needed to carry out the operation. It does not do a good job, however, in predicting when events will actually occur.

The relevance of this discussion is that a potential future direction for SAIL is to follow up the deterministic scheduling of ships with a stochastic simulation of the realizations of that schedule, and to report the expected results more realistically.

3.3 INTERFACES OF SEALIFT PLANNING WITH OTHER MEMBERS OF THE JOINT DEPLOYMENT COMMUNITY

Clearly, all organizations involved in a military operation must work together to create a consistent plan. Each command, however, has a few other commands with which it is most closely allied. For sealift planning, these closest associates are the supported commander, who is in charge of conducting the military operation, and the Military Traffic Management Command, which manages land movements in the United States and which operates various port facilities worldwide.

The primary objective of the materiel movers is to satisfy the requirements of the supported commander. Planning permits these movers--MSC, MTMC, and MAC--to determine the resources required for supporting the operation, to assess the feasibility of the plan, and to alert the supported commander concerning any transportation shortfalls.

In the other direction, MSC must work closely with MTMC, which is primarily responsible for the management of transportation resources for moving personnel and materiel from origins to the ports of embarkation. The choice of ports to be used and the timing of the arrival of both ships and materiel at the ports are two important decisions that must be resolved between these two commands. A concept is currently being developed that will link the planning systems of the commands; this should result in improved resource utilization and more timely deliveries of cargos. More information about this interface can be found in the ORNL draft document, *A Proposed Concept for the MTMC/MSC Interface.*

In addition to the close interaction with the supported commander and MTMC, the sealift planners must keep the remainder of the planning community informed of their activities. Reporting plan changes to a central information file is expected to be a continuing part of any planning system. In recent years, the Joint Deployment System has been the focal point for accumulating this information.

3.4 INPUTS TO THE SEALIFT PLANNING SYSTEM

Let us turn now to the kinds of information required by the sealift planner to carry out MSC's planning objectives. The purpose of this section is to describe the types of available data in terms that may be helpful in understanding the SAIL model; specific origins of the data and detailed descriptions of data content are given in Section 5. The data fall into

four main categories: information related to cargos to be carried, ships available for use, the ports and other resources involved, and timing information.

3.4.1 Cargo Data

Fundamentally, the information required concerning the items to be shipped are (1) who the shipper is, (2) what the cargo is (by category code), (3) when and where it is available to be shipped, and (4) when and where it is needed. There are other attributes that may also be important, such as whether or not the cargo must make an intermediate stop, and special handling considerations. Often, an objective of movement planning is to transport all materiel for a military unit in a single shipment, if possible. In contrast to unit cargo, nonunit cargo is general resupply materiel, not associated with a particular unit.

There is some additional information known about cargos and ships which impacts the way in which resources can be used. The rate at which each class of cargo can be loaded and unloaded from the various compartments of different types of ships is given. Also, there is a stow factor for all combinations of cargo classes and compartments. This factor expresses the efficiency with which a cargo can be stowed; a stow factor of 80%, for example, indicates that 100 measurement tons of cargo will actually use 125 measurement tons of space. Finally, there are some combinations of cargo classes, ship types, and port facilities that are favorable, other combinations that are feasible but less favorable, and some combinations that are infeasible.

The amount of detail with which a planning system deals is an important modeling consideration. Too much detail will swamp the system with special cases and decision rules. Too little information will result in important constraints being violated, which in turn diminishes the usefulness of the plan.

3.4.2 Ship Data

The planning system must know (1) the ships that are available for executing the plan and (2) certain attributes concerning the ships. The type of ship (which indicates the kinds of materiel it can carry), capacities of the ship's compartments, nominal speed, and information about when and where the ship will become available are key elements. Also of importance is the fleet to which the ship belongs (which may impact how the ship can be used), its dimensions (which affects which ports and canals may be accessed), and special capabilities the ship may have (such as boom capacities). A minimum utilization percentage may be specified for ships; this prescribes that a ship must be carrying at least that fraction of a full load before the ship is permitted to sail.

3.4.3 Port and Other Facility Data

The most important elements in this category are the location, capacities, and physical characteristics of the ports involved in the plan. In this report, the word port is used to identify places with unique geolocation codes. Facility is used for each of several types of cargo-handling capabilities at the port: general, ammunition, container, and petroleum. The maximum daily throughput of each facility is specified. Some types of cargo may be excluded from some ports, either for policy reasons, or because the port lacks the facilities to process

that type of cargo. Particular cargos may pass through the port, but the type of ship may be restricted; for example, ports lacking container-handling facilities can load and unload containers, but only on self-sustaining ships.

Other important facility resources are the Panama and Suez canals. The size of ships they can accommodate and their assumed availability throughout the plan are important data elements.

3.4.4 Distance Computation Data

To support the computation of the distances between the ports involved in a given plan under varying combinations of canal closings, an input file of global positions must be available. As implemented in the SAIL algorithm, this takes the form of a set of nodes, or positions, and a corresponding set of links which describe the connections among the nodes. This set of nodes and links forms a network on which the ships may travel. The number of nodes must be sufficient to outline coastlines and to allow ports and ship positions to be attached to the network with adequate accuracy in the distance calculations. These distance files are not expected to change frequently. Special at-sea locations or a new port in a remote part of the world may require the addition of new nodes and links, but in general these files should be of no concern to the planning analyst.

3.4.5 Timing Information

A variety of time-related variables are needed by the system; these include the availability dates of ships and ports, dates on which the capacities of ports may change, and the timing related to cargo movements. Other special-purpose, time-related variables will be introduced in the sections that contain the descriptions of the model and the computer program.

3.4.6 Other Data

A variety of other data supports special features of sealift planning, such as ship attrition, planner-specified loadings, and other restrictions and opportunities concerning cargo routing or handling. The features are given in Paragraph 4.3, and the data elements are described in Paragraph 5.2.3.

3.5 THE RESULTS OF THE SEALIFT PLANNING MODEL

The next several paragraphs outline the outputs that the scheduling system returns to the database management portion of SEASTRAT. Note that the results of the scheduler are not designed for direct presentation to the planning analyst; the data management and report generation activities of SEASTRAT must intervene to permit the following "raw" schedules to be more easily read and interpreted.

3.5.1 The Assets Required

One of the more fundamental questions in the planning function concerns the number and type of shipping resources that are required to support the movement requirements. As seen in Paragraph 3.4.2, the SAIL algorithm expects a list of shipping resources available as an input. Typically, not all of these resources will be required. In fact, a subobjective of the model is to limit the number of ships used. Further, the minimum load requirements will prevent the use of sparsely loaded vessels. The scheduler provides a file of the ships used, the ports visited by each, distances sailed, and the estimated number of bunkering (refueling) stops made.

3.5.2 The Deployment Schedule

The central output file is a detailed itinerary for each ship, giving the ports visited, the cargos loaded or unloaded at each port, and the time of each visit. It is from this file that most of the reports of interest to the planning analyst are developed. The timeliness of the scheduled delivery of cargo, scheduled closure dates of units (the time all cargo for a unit is delivered), miles traveled, and the effective utilization of ships can all be assessed from this file.

3.5.3 Analysis of Alternatives

It is expected that the scheduler will be used to evaluate the impact of varying assumptions on the quality of the schedule produced. For example, the analyst may wish to determine the impact of ship attrition, canal closings, or varying ship availability on the schedule's quality. The deployment schedule file remains the fundamental instrument for these assessments. However, it is difficult to compare two of these large data sets for overall quality. To aid in these comparisons, a variety of statistics are developed; for example, histograms of cargo quantities delivered on time, days early, or days late are drawn for various accumulations of cargos. (For example, a histogram may be useful for each cargo class, for unit and non-unit cargo, or for all cargo due for delivery within a particular time interval.) Also, the scheduling system can compute summary statistics of on-time, late, and undelivered cargos, summarize early and late delivery tonnages, and show the number of each type of ship used along with the number of trips, miles sailed, and average utilization. Simple indices of plan quality have been examined but have been found to have little utility for comparing the results.

4. THE SEALIFT SCHEDULING MODEL

In the previous section the sealift problem was defined, emphasizing the data going into the model and the results expected from the model. In this section, the way in which those results are obtained is discussed. To begin with, some concepts used in the SAIL model are reviewed. Then, an overview of the approach is given; and finally, a number of special scheduling concerns will be defined and the way in which the model addresses those concerns is explained.

4.1 THE SCHEDULING CONCEPT USED IN SAIL

The sealift scheduling problem, like many transportation-related problems, is difficult to rigorously formulate, and even more difficult to solve. Building a mathematical representation of the problem is complicated by the often obscure nature of the objective. For example, the primary objective may be to deliver all cargo as close to "on-time" as possible...but, using as few ships as possible, loading all ships as "full as possible," keeping units together "as much as possible," etc. The multitude of subobjectives almost certainly conflict with each other and will generally conflict with the primary objective.

Even if the problem is unambiguously formulated, it is beyond current technology to solve. In rather formal terms it is described as a very large nonlinear optimization problem. Generally, problems of this type are not amenable to exact solution, and the sealift scheduling problem is no exception. The number of combinations of ways to route ships and load cargos is astronomical; finding the true optimum solution is generally not possible.

Given these difficulties in both formulating and solving the sealift problem, the SAIL algorithm has been designed to accept varying objectives and, having formulated the problem, find a "good" solution. As the following examples show, optimization methods, heuristics (rules), and simulation are all used along the way to obtaining that solution.

Linear optimization is used to make optimal assignments of cargos to ships, once the routes of the ships have been established. The primary objective is to minimize late deliveries; the constraints are the shipping capacities between origins and destinations as defined by the predetermined ship routes. This method selects the "best" assignment of cargos to ships from among the many possible choices. In addition to finding an answer that is best (with respect to a specific objective), these optimization tools are efficient ways of sorting through the large number of combinations of potential assignments.

Implementing sets of decision rules on a computer is probably the most common modeling technique. The method is flexible, capable of addressing complex situations, and deals in logic that is familiar to the planning analyst. Processes involving very carefully structured decision rules are enjoying a renaissance under the title of artificial intelligence. The SAIL model employs rules at many points in its analysis, as will become evident in some of the following paragraphs. One use will be highlighted here. In one version of the ship

routing algorithm, linear optimization is used to find good candidate cargos for a ship to carry. One difficulty with this approach is that, when adding a cargo to a ship, a cargo may be a good candidate by itself, but may be a poor candidate in combination with other cargos. This led to the development of a hybrid process, interleaving optimization with heuristics. The idea is to allow the optimization to find candidate choices from among the large number of possibilities; then, rules can be employed to resolve the conflicts which arise within the optimization results due to the interdependencies. A more restricted optimization problem is then formulated, and the process is repeated. This continues until the problem is solved.

Simulation is another common modeling method. Essentially, this process involves programming the computer to duplicate the system under study. Simulation, therefore, not only incorporates the decision rules, but also estimates what will happen to the system as a consequence of making a particular decision. In some simulations, both the decisions and their consequences may be affected by randomness, which allows the model to recognize the unexpected or unpredictable elements in the system. These simulations, sometimes called stochastic simulations, are repeated a sufficient number of times to permit statistics to be gathered; the results are then presented as probabilities that certain events will occur.

SAIL does not now use stochastic simulation. Rather, simulation is employed to add detail to the somewhat bare results of the optimization and heuristic processes. Most importantly, the times at which events occur are estimated throughout the analysis; simulation is used to refine these time estimates by stepping through the implementation of the decisions. It is important to note that random effects are not considered; the time it takes a ship to sail, for example, is based on the ship's speed as contained in the ship file. Variations due to weather or operational concerns are not considered.

A variety of strategies can be used to interface the transportation analyst with the modeling system. At the simplest level, the analyst sets certain parameters that govern the behavior of the system and then has no further interaction with the model until results are presented. More complex systems may permit the user to interact with the system, altering the behavior of the model as it proceeds toward a solution. At this writing, SAIL operates at the simpler level; there are a number of parameters which affect both the optimization and decision rule processes. True interaction, however, is not expected to be incorporated until Execution Planning is implemented.

4.2 AN OUTLINE OF THE SOLUTION PROCEDURE

In this section, the major steps by which the SAIL model obtains a solution from the input data are described. Appendix A gives an outline of this process. The steps are currently executed serially; that is, each step is processed in turn, and there is no looping back to earlier steps.

4.2.1 The Database Management System Interface

As would be expected, the first step is to link the database management system to the SAIL module. This involves structuring the required data in such a way that it can be conveniently processed by the model. Some editing of the data is done at this time, but the

SAIL model expects most data errors and inconsistencies to be resolved by the database management system. The interface process is explained in Section 5.1.2.

4.2.2 Cargo Aggregation

Deployment data identify cargo in too much detail to be feasibly processed in the transportation planning model. Individual cargos are therefore aggregated into larger entities called "channels." A channel will contain all cargo that is of the same cargo class, involves the same ports, and that needs to be shipped at about the same time. (The analyst can control the variation in time that is acceptable for aggregating cargos into a channel.) In addition, special characteristics of the cargo, such as intermediate stops, Assault Follow-on Echelon (AFOE) and Consolidation of Load (CONSOL), and unit information is used in the aggregation process. A complete list of aggregation criteria is given in Paragraph 5.1.3.6.

For each channel, a nominal shipping date is computed. This date represents the starting date required for a ship of average speed to load the cargo at the port of embarkation, sail to the destination, unload at the port of debarkation, and meet the required delivery date. The nominal shipping date cannot be earlier than the available to load date (ALD) of any of the cargos in the channel.

In addition to channels, a less restrictive aggregation of cargo is also made. These entities, called "clusters," are essentially aggregations of compatible channels. Clusters are made up of channels that have similar ports and time requirements but may have different cargo classes. Ammunition and AFOE cargo will not be clustered with other types of cargo.

The use of channels and clusters will be discussed in detail in Paragraph 4.2.5. Basically, channels are the finest definition of an aggregation of cargo required to make an assignment to a specific ship compartment. Clusters, on the other hand, are groups of channels that could sensibly be carried together on a ship, if the ship is able to accommodate the cargo classes involved.

4.2.3 Distance Computations

Three files are of central importance to the distance computations in SAIL: a list of nodes, or points, located strategically throughout the oceans; a list of links, which define the connections between the nodes; and a list of "lines," which cannot be crossed by ships. Figure 4.1 shows a sample of an ocean region with its nodes, links, and lines. Each of these entities will now be discussed in more detail.

Nodes are defined by their latitudes and longitudes; each node is identified by a unique name. Nodes are placed throughout the ocean with several notions in mind. First, as can be seen in Fig. 4.1, all major coastlines are roughly outlined by a series of nodes. Also, there should be a node at or near all important ports and nodes along major shipping routes. Finally, for reasons that will be explained shortly, there should be no ocean region that is too far from any node. (About a one day sailing distance, roughly 400 nautical miles, is currently considered too far.)

ORNL DWG 88-340

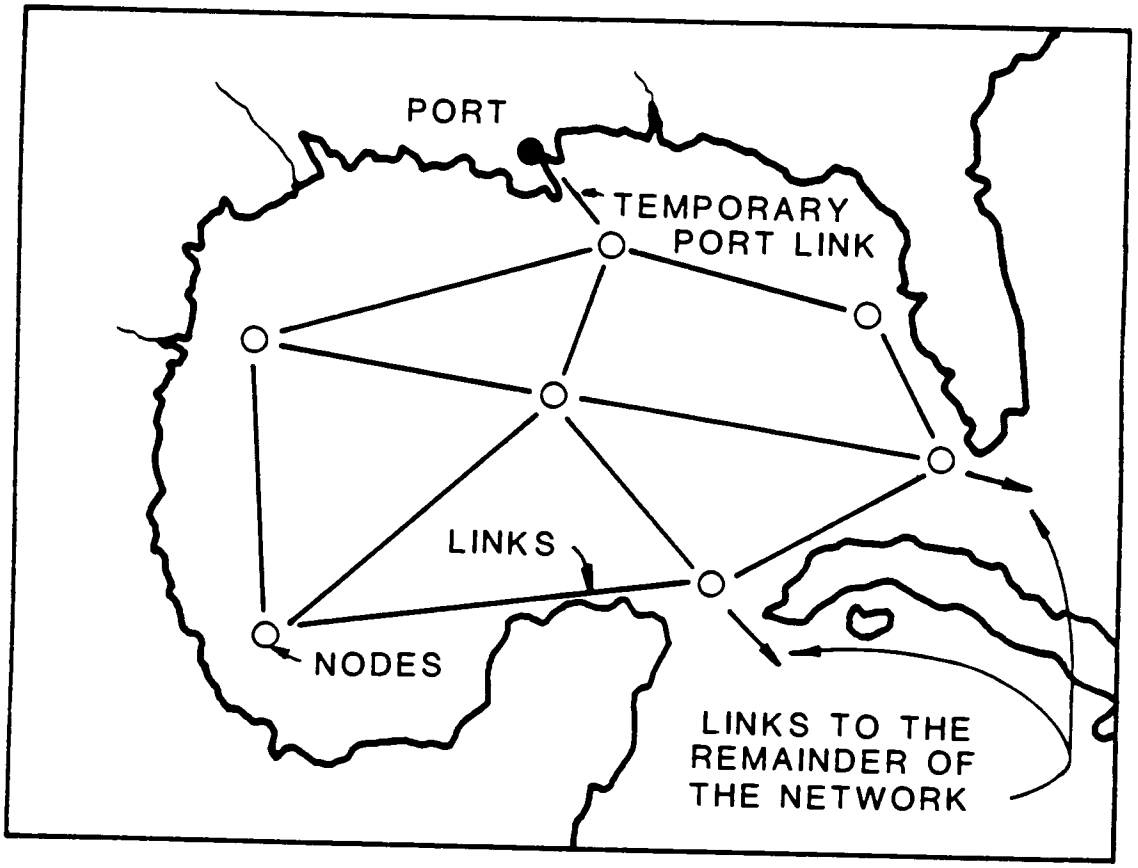


Fig. 4.1. Part of an ocean network.

Each link is identified by a name and has three parameters: the nodes at each end of the link, and the distance between the nodes. The distance may be input to the system, or, if it is missing, the system will compute the great circle distance between the nodes and put this calculated distance in the link file. The links, then, form the paths over which ships may travel. The nodes and links therefore need to be laid out in such a way that all major shipping routes are well defined and all ocean regions are well-covered. As a matter of definition, this grid of nodes and links is called the ocean network.

Before describing the third of these entities, lines, it will be helpful to discuss the way in which the distance subsystem is used. The most fundamental requirement is to be able to compute a set of distances between all pairs of ports and at-sea locations involved in a deployment. To do this, each port must be connected to the network. If a node is at (or sufficiently near) the port location, the port is already on the network; otherwise, a new, temporary node is placed at the port, and a link established to the nearest existing network node.

After all ports are linked to the network, the distances between all pairs of ports are computed using a shortest path algorithm. (This is a common computational process in operations research; the interested reader may wish to look at Chapter 7 of Wagner.) These interport distances are found for all combinations of the Panama and Suez canals being open or closed.

Note that there is a tradeoff between accuracy and computation time. A denser network can represent routes more precisely; however, the marginal improvements will ultimately be outweighed by the computation time, which increases nonlinearly with the number of nodes.

In addition to ports, the ships that may be used in the deployment must be connected to the network. A process similar to that used for ports is used, except that there is no temporary node placed at the ship's initial position. This process is used for the calculation of the availability date of ships at the various ports.

This brings us back to the third entity: lines. A line is defined by the location of its two end points and represents a region that cannot be crossed when linking either ports or ships to the network. Lines are useful on narrow strips of land where the node density might be insufficient to adequately define the coastline. (Lines are also likely to be used in future work where particular areas of the ocean may be treated specially: areas to be avoided or areas of high risk, for example.)

A table of distances among ports is computed which gives the nautical miles from each port to each other port assuming the two major canals are closed. The distance from each port to each canal is also found. When a specific port-to-port distance is required in the SAIL algorithm, the shortest route is found from the table for the prevailing canal conditions. (For efficiency, pairs of ports that do not use canals are marked, obviating the need to find the shortest path.)

It is also possible that the paths ships may take will vary through time. For example, far northern latitudes may be impassible during winter months, or some regions may be

excluded from travel at particular times due to strategic or political considerations. This capability is accomplished by using the Ocean Avoidance concept (also referred to as Exclusion Areas). Clearly, the distance between ports may change as these exclusion areas change; a different distance table is then required for each time exclusion areas change. The general distance calculator is therefore a function of the ports involved and the time of the voyage.

4.2.4 Ship Availability Dates

An initial availability date for each ship at each port in the deployment plan is computed using the ship's starting location, current destination (if any), and a series of rules which are dependent on the ship's fleet and whether or not wartime conditions prevail. Also, certain delays or accelerations (early warnings) in availability may be specified. Basically, under peacetime conditions the ship will be allowed to proceed to its original destination; after adding and subtracting time for delays and early warnings, it may then proceed to a port of embarkation. Under wartime conditions, the ship bypasses its original destination. Appendix H gives the details of these decision rules.

4.2.5 The Ship Routing Subsystem

At this point, the size and nature of the deployment planning problem presents us with a dilemma: on one hand, the routing and scheduling of ships are such difficult problems that they need to be separated; on the other hand, the routing of ships cannot really be done properly without a consideration of the eventual scheduling of cargo on those ships. The SAIL model resolves this conflict by developing routes with tentative schedules, followed by a separate step that refines the loading of cargos on the routed ships.

Let us review what is known at this point in the execution. There is a set of aggregated cargos, called channels, with the following known characteristics:

1. ports of embarkation and debarkation,
2. time of cargo availability,
3. latest desirable delivery date,
4. an estimated shipping interval, and
5. the type of cargo in the channel (which governs the type of ship required).

The characteristics of ships, including their speed and capacities, are known, and the date when each ship will be initially available in each port has been calculated. The types of cargo that each port can process and the throughput capacities are known. Interport distances have been computed for all combinations of canal states. Load and unload rates, stow factors, and allowable combinations of ships, cargo classes, and port facilities are known. Further, ship dimensions can be used to determine if ports and canals can accommodate the various ships.

From this information, a problem can be constructed that has as its objective the matching of channels to ship sailings. Two techniques have been used. The first method uses

optimization to select candidate assignments for ships; it then uses decision rules to test each candidate for inclusion or exclusion. The second method is totally heuristic, or rule-based.

In limited testing to date, the optimization-based method gives slightly better answers in most cases, particularly when shipping resources are scarce. It does, however, require much more computation time which may render it unacceptable for very large problems. Both of these methods are retained in the present SAIL version; a parameter controls which is used in any particular case.

4.2.5.1 Optimization-Based Routing

This option uses the standard transportation problem formulation, which is a special case of linear programming. Both of these terms are common in operations research literature; details may be found in Wagner, Chapters 2 and 6. A nontechnical introduction to linear programming is given in Appendix B, and a formal statement of the routing problem formulation is given in Appendix C.

Note that petroleum shipments are formulated in a problem separate from dry cargo and passengers. This is possible because no resources are shared between these two major cargo types. This partitioning results in reduced computation times.

The transportation problem is formed by a set of demands (which in our problem are the channels) and a set of supplies (which in this case are shipping resources). Both the demands and the supplies have known capacities, and in this formulation, the total of all demands must equal the total of the supplies. If that equality is not inherent in the problem, as is generally the case, a "dummy" supply or demand is added to balance the total quantities. It should be mentioned here that both the shipping resources and the demands are adjusted to approximate the effect of stow factors.

Finally, there is a cost, or penalty, associated with assigning a unit of any supply to a unit of any demand. A very large cost is placed on any disallowed combinations. The objective is to assign supplies to the demands in such a way that the total of all costs are minimized.

In our problem the demands are aggregations of cargos (channels); the supplies are ship compartments. Even though each ship can make more than one trip, it is only the most imminent trip of each ship that is explicitly represented. All subsequent trips of all ships are represented by a single large capacity, called the dummy ship. As will be seen below, channels that cannot be shipped for some reason will remain on the dummy ship at the end of the routing process.

There is also a dummy channel, which represents unused space on the ships.

The cost associated with any channel-capacity pair is a function of the various objectives of the planning analyst. On-time delivery of cargo is generally of first importance; avoiding lateness will thus dominate the cost function. Other objectives are limiting early delivery, reducing the number of ships used and miles sailed, and achieving desirable matches of cargo classes and ship types. Note that placing cargo on the dummy ship (which is

equivalent to not shipping the cargo) carries a high penalty. Cargo-ship matches that are not feasible (such as placing materiel on a passenger ship) have a prohibitively high cost. Clearly, there is no penalty for placing the dummy channel on the dummy ship. More will be said about how the cost function is manipulated during execution of the routing system as the routing concept is developed in detail.

The mechanism for establishing whether or not a ship will be able to deliver a cargo on time is derived from the estimated shipping date computed when the channels were formed (see Paragraph 4.2.2). The comparison of this date with the ship's availability date in the port of embarkation gives an estimate of the timeliness of delivery for that combination of channel and ship. (Note that the estimate is based on the ship carrying only the single channel; extra delays may result if the ship services other channels as well. This issue will be considered in more detail below.)

The transportation problem is now completely formulated; a solution to the problem is now found using a tailored version of a standard algorithm for finding a minimum-cost solution (see Wagner, Chapter 6). This solution gives the amount of each channel to place in specific compartments of each trip of each ship.

The immediate problem with the solution is that incompatible channels may be placed on a ship. For example, the solution may prescribe that the first trip of a ship should take some cargo from San Francisco to Honolulu and some other cargo from New York to London. This may occur since the linear model does not consider the interactions among the variables; that is, how the decision to load one channel on a ship might affect the cost of loading another channel on the same ship. Each of these cargos was considered to be independent of all other cargos, and the costs were computed under that assumption.

The procedure used to resolve these inconsistencies in assignments is now outlined. First, a dominant, or prime channel assignment is found for the first trip of each ship. Dominance is a trade off between the amount of cargo assigned and the time the cargo needs to be shipped. A decision rule is used to find a suitably early, large assignment.

After the prime assignment is found for each ship, the ship with the largest such assignment is selected for sailing. The first step is to eliminate channels that are geographically incompatible with the prime assignment. If there are any such channels, their cost is raised and the transportation problem reoptimized. (Note that, for computing efficiency, reoptimization always begins with the previous optimal answer.)

Following the reoptimization, the chosen ship will be loaded only with geographically compatible channels. (Note that this reoptimization may attract channels which were previously assigned to another ship.) Time-related compatibility must now be established. This is done by simulating the trip with only the prime channel and noting the delivery time. Other channels that the solution still places on the ship are then added to the trip, one at a time, checking their delivery dates and their affect on the delivery dates of all other channels. Only those that enhance the overall delivery effectiveness of the trip are retained. As each successful candidate is added to the ship, the amount of stowage used is computed using the actual (rather than estimated) stow factors.

The current trip of the selected ship is now sailed. Two things happen as a result of this sailing. First, the cargo quantities carried by the ship are removed from the transportation problem; and secondly, knowing the route the ship will take allows the ship's availability dates for its next trip to be computed. From these dates, new costs for that ship's next trip are found. At this point it should be noted that the first trip of all ships carry a cost penalty for the initial use of the ship; this strategy tends to minimize the number of ships used. Once the ship has sailed, this penalty is removed for subsequent trips.

The problem is then reoptimized, and the process of finding a new prime assignment begins again. This procedure continues until all channels are completely assigned to ships that have sailed, or are forced onto the dummy ship by the inability of the procedure to find a compatible assignment on a real ship.

Finally, the representation of stow factors in the transportation problem should be addressed. As was described in Paragraph 3.4.1, the stow factor is the stowage efficiency of a class of cargo in a particular type of ship compartment. However, the special structure of the transportation problem does not permit a different factor for each channel-compartment combination. Instead, a vector of factors is established for the ship capacities, and another vector of factors is applied to the channels. These vectors are constructed such that their product yields a matrix which approximates the true matrix of stow factors. (The last column and last row of the stow factor table in Appendix E are the ship and cargo vectors respectively.) These vectors are then used to increase the apparent quantities to be shipped and decrease the apparent shipping capacities in the transportation problem. The result is an assignment of cargo to ships which takes stow factors into account. The method will, however, overestimate the stow factor in some cases, and underestimate it in others. But, as the channels are evaluated for placement on the ship by the decision rules, the true stow factors are used. Thus, although the optimization portion of the routing system uses approximations, the final result is ships loaded in accordance with correct stow factors.

4.2.5.2 The Heuristic Routing Approach

The notion behind the previous routing process was to simultaneously evaluate large numbers of potential assignments of channels to ships and use optimization to select the best set of candidate assignments. The quality of answers obtained by this approach appears to be good; computation time is another matter. Even after extensive efforts to improve the solution efficiency, the time required to obtain answers to large deployment problems is likely to be unacceptably high.

For this reason, a purely heuristic approach to the routing problem was developed as a switch-selectable option. The method uses the same structure as the optimization version: the problem is set up in precisely the same way, including the formation of the cost matrix. But the optimization problem is never solved. Instead, a decision strategy is employed: note that different strategies can be inserted, and several have been tried so far.

The current approach attempts to assemble a good shipload of cargos, including the next most critical channel (the channel that, by some set of criteria, most urgently needs to move). This involves the following steps:

1. Channels that still have cargo to move are evaluated on both urgency of the movement and the quantity of materiel to move to determine the most critical channel. (This will be referred to as the prime channel.)

2. All channels clustered with the prime channel are marked. The idea is that clustered channels may form the basis for a good ship load. (Recall that clustering is based on origins, destinations, and approximate shipping times.)

3. Using the cost function as a guide, the best ship for carrying the prime channel is identified.

4. All ships that have costs within some tolerance of the best ship are marked for consideration.

5. The list of channels from step 2 are trial loaded onto each of the candidate ships from Step 4. The effectiveness of each load is computed based on the amount of the channels carried and the amount of empty space left on the ship. The ship with the best score is selected for the trip.

6. The prime channel is loaded onto the selected ship. Other channels that are compatible with the prime and that can be successfully delivered by the ship are then added, subject to space availability, loading constraints, and impact on the delivery of channels already on board.

7. The ship is then sailed, and the process begins again, continuing until all cargos are moved or it is determined that they cannot be moved.

4.2.5.3 Ship Routing Summary

Let us now summarize and review the product of the Ship Routing Subsystem. By one of two methods a prospective set of channels to be carried on all trips of each ship has been established. This prospective list of channels to be carried has two important implications: (1) it establishes a tentative list of ships to be used in the deployment, and (2) a route for each of the ships to follow is implied by the channels placed on them.

4.2.6 The Ship Loading Subsystem

The objective of the Ship Loading Subsystem is to more effectively allocate cargos to ships now that routes for all ships have been established. Two steps are involved in accomplishing this objective. First, simulation is used to obtain accurate estimates about the time each ship departs each port on its route. (This step uses the Simulation Subsystem described in Paragraph 4.2.7.) Then, costs of lateness or earliness can be found for all feasible combinations of channels and ship trips. A new optimization problem is then constructed; because the routes are known, the number of variables in the problem is much reduced from the ship routing formulation. Also, unlike the routing problem, the loading problem can be formed as a single optimization problem; since the routes are established, it does not require the intervention of rules to resolve conflicts.

The structure of the problem is that of a general network problem, which is another variation of linear programming. Throughout the remainder of this topic, the reader will need some basic understanding of linear programming problem formulation. The reference in Wagner, Chapter 3, is a recommended introduction.

The reason linear programming was not used initially is that the problem size would have been too large for effective computation. Each feasible combination of channels and compartments would have been a separate linear programming variable; further, the constraints would have included all possible combinations of ships, trips, and compartments. However, after the Routing Subsystem has finished, the number of variables and constraints has been reduced significantly by eliminating trips that did not appear in the transportation problem solution. Also a route was established for those ships that were in the solution, which in turn limits the potential cargo assignments for each ship.

A formal statement of the linear programming problem appears in Appendix D. The problem is formed around the following ideas. First, each potential assignment of a channel to a compartment, as indicated from the routing solution, becomes a variable in the linear program. This variable represents the amount of the channel which will be carried. All feasible combinations of channels and ships will be in direct competition on the basis of cost. The objective will be to deliver all of the cargos at the minimum total cost.

But what about channels that cannot be delivered? Once again, a dummy capacity must be introduced; each channel has the opportunity of being placed on this dead-end, but at a high cost. This assures that non-delivery is the last resort for all cargos.

There are two kinds of constraints: the first requires that the total quantity of each channel must be placed on a real ship or the dummy ship, while the second type of constraint limits the *stow-adjusted* quantity of channels in a compartment to the compartment's capacity.

A mathematical formulation of the loading problem is given in Appendix C.

The result of the Ship Loading Subsystem is a set of stow-adjusted assignments of channels to routed ships which maximizes on-time deliveries, given the ship routes established in the Ship Routing Subsystem. This solution will now be passed to the Simulation Subsystem for the addition of detail and the refinement of the timing of events.

4.2.7 Simulation Subsystem

Of the major computational subsystems in SAIL, this is the simplest in concept and most intricate in implementation. At this time we will deal primarily with the concept, leaving the details to Paragraph 5.1.7 which deals with the software.

There are seven principal tasks reserved for the simulator:

1. refining the time at which events occur;
2. refining the sequence of port visits on a route;
3. disaggregating channels back into individual cargos, as defined by lines on the cargo data file;

4. de-scheduling cargos that would be delivered unacceptably late;
5. removing from the schedule ships that are too lightly loaded;
6. removing port stops for which there is too little activity; and
7. adding bunker stops, as needed, if the fuel computation option is selected.

The basic approach is as follows.

First, the solution from the optimization is organized into a comprehensive list of tasks for each ship to perform. From this list, an efficient sequence of port visits is determined for all trips, resulting in a time-ordered list of activities for each ship. The time that a ship may begin its first activity, which is always a loading operation, is found; this is usually the ship's availability date at the first port, but it may be delayed due to cargo or port restrictions. In addition to loading, there are activities of unloading, sailing, and docking.

The simulation proceeds by choosing the ship with the next occurring activity. The time required to perform the function is computed; this time is added to the start time of the activity, giving the starting time of the ship's next function. This continues until all activities have been processed.

As we have seen, the optimization processes deal only with channels. The loading activities are described in terms of these aggregations of cargo. During the simulation, particular cargos contained in the channel are picked for loading; the order of choosing these cargos is based mainly on required delivery dates and cargo availability dates.

Removing unacceptably late cargos and lightly loaded ships from the schedule involves multiple passes through the simulation procedure. On the first pass, cargos that are delivered after their last acceptable date are marked; on the next pass, there is an attempt made to accelerate the loading and delivery of these marked cargos. However, if they are still late, they are removed from the list of cargos to be delivered. Subsequent passes then develop the schedule without the presence of the deleted materiel. (Note that in cases where cargos are carried by more than one ship, only part of a cargo may be deleted.) Finally, after all unacceptably late cargo is removed, the ships are checked to see if they meet minimum load standards; if not, the trip is removed from the schedule.

The end product of the simulation is a list of all load and unload activities of each ship. For each such activity, the identification of the cargo involved, the amount of cargo, and the starting and ending times are given. Another list gives the ship's itinerary, the number of miles sailed, and the number of bunkering stops required.

4.2.8 The Reporting Subsystem

The information developed by the Simulation Subsystem requires some enhancement before it can be conveniently accepted by the database management system. Most importantly, in keeping with the overall goal of avoiding late deliveries, the simulation places events at the earliest possible time. This procedure results in ships arriving in port, and some cargos being delivered, unnecessarily early. The Reporting Subsystem moves these events to later times whenever it is possible to do so without creating additional lateness in the schedule.

This subsystem also develops summary statistics of the schedule. These numbers are intended to facilitate comparisons of quality among differing schedules of the same plan. (For example, a plan may be considered with several different sets of ships starting at different initial locations. The variance in the quality of solutions obtained under these different starting conditions may be of interest.) The type of summaries computed include histograms of earliness or lateness of delivery by cargo class, percentages of on time, late, and undelivered cargo, ship utilization, and port activity.

The principal product of this subsystem, and hence, the principal product of the SAIL model, is an itinerary of each ship's activities, including a range of dates at which the activity may occur. This final output file is described in detail in Paragraph 5.2.4.2.

4.3 SPECIAL SCHEDULING CONCERNS AND PROBLEM FEATURES

To prevent the overall description of the SAIL model in Section 4.2 from becoming too intricate, several special concerns of the sealift analyst were avoided. In this section, these features will be introduced and their basic impact on the model will be indicated. Most of the features concern how and where ships can be loaded, special restrictions on the way they can be routed, and what may occur to them along the way. The order of presentation follows, as much as possible, the chronological order of events during a shipment. Certain features will be implemented in the future, and are so indicated.

4.3.1 Fleet Use Restrictions

Some fleets of ships may be restricted to use in certain areas of the world. For example, Korean ships may only be used in the defense of Korea. This restriction imposes an additional reason for limiting cargo items to particular ships, a feature that is already present in the system for other reasons.

4.3.2 Ship Size Restrictions

Ships are excluded from using port facilities or canals due to size limitations (length, beam, and draft). A comparison of ship size and facility accommodations is made at the time costs are computed for the transportation problem formulation. Infeasible loadings due to size are given a prohibitively high cost.

4.3.3 Container Ship Facility Requirements

Non-self-sustaining container ships are restricted to ports that have container facilities. Again, high costs in the Ship Routing Subsystem are used to restrict the routing of such ships.

4.3.4 Variable Draft (Future)

By using partial loading, a ship may be able to enter shallow-draft ports that it would not be able to visit if fully loaded. A future development is to exploit this idea to expedite deliveries at such ports. The problem is complex and has not been worked out at this time; one possible approach is to modify the stow factor for cargos bound for shallow ports.

4.3.5 Queuing at Ports

The throughput capacity of a port facility limits the number of ships that can be served, potentially causing queues of ships to develop. The Ship Routing Subsystem and the Ship Loading Subsystems estimate the length of such queues and consider the resulting time delays in the cargo delivery computations. The Simulation Subsystem explicitly models the activity at the port, including the waiting time. Switches are provided to allow queuing delays to be omitted at any port at the planning analyst's discretion.

4.3.6 Port Availability Dates

An earliest availability date may be specified for each port. Presumably, cargo will not be required at ports prior to that date; the impact is primarily one of limiting the early delivery of cargo. This is monitored in the route simulation portion of the Ship Routing Subsystem; cargos that would detrimentally delay a ship due to a port being closed are excluded. Naturally, this feature is also modeled in the Simulation Subsystem.

4.3.7 Port Facility Throughput Changes

The capacity of ports may change through time, perhaps due to the arrival of special-purpose ships or other port improvements. Port throughput will then be a function of time; this factor will impact queuing delays over the course of the plan.

4.3.8 Preferred Ship Loading

The planning analyst may wish to specify that a cargo be placed on a particular ship. One reason for doing this is to properly load and schedule special cargos that might not be handled correctly by the model. In the Routing and Loading Subsystems, the cost functions are used to implement this feature: the cost of placing the cargo on the specified ship will be low; the cost of placing it on any other ship will be prohibitive. It is also possible to restrict the loading of a cargo to a particular type of ship.

4.3.9 Flexible Cargo Loading

Cargo does not necessarily have to be loaded on the most suitable type of ship. For example, vehicles can be loaded into the holds of breakbulk ships, and evacuees can be carried by almost any kind of dry-cargo ship. On-time deliveries are generally aided by this kind of flexibility, but at some cost. Loading and unloading rates may be longer or the space utilization may be less efficient when cargo is placed on less suitable ships. Although less-than-ideal loadings are permitted, they are discouraged by the cost function.

4.3.10 Unit Integrity (Mainly Future)

It is desirable to keep all elements of a military unit together during shipment. Depending on the nature of the cargo and the type of ships available, this goal may or may not be feasible. The current system aggregates cargos by unit; both the Routing and Loading algorithms then tend to schedule all of a channel on a single ship, if possible. However,

different classes of cargo for a unit will still be in separate channels, which may be carried on different ships.

This issue will likely be investigated more thoroughly. The trade-offs between unit integrity and other objectives, such as the efficient use of lift resources, will need to be well defined before undertaking the effort.

4.3.11 Ammunition Routing Restrictions

As a general rule, a ship carrying ammunition cannot enter a non-ammunition port facility. Therefore, if ammunition is mixed on a ship with other cargo, the ammunition must be loaded last and unloaded first. Logic to assure the constraint is honored complicates both the Ship Routing and Simulation Subsystems.

4.3.12 Changes in Canal States

The planning analyst can specify that either or both the Suez and Panama canals open or close at varying times through the planning period. As was mentioned in Paragraph 4.2.3, distances among ports are computed for all combinations of canal states. At any time during the schedule, the set of distances which properly reflect the canal states will be used. It is assumed in this process that the ship will have prior knowledge of the canal status and take the shortest appropriate route.

In the future, the assumption of prior knowledge of canal states may be removed. The ship will sail expecting the canal to remain in its current status; if the status changes while in route, the ship's route may need to be adjusted.

4.3.13 Delays at Canals

The time required for a ship to pass through either canal is specified. This time, which may be in fractions of days, is added to the duration of any trip using the canal.

4.3.14 Intermediate Stops

Some cargo makes an intermediate stop between its ports of embarkation and debarkation, and may be required to remain at the intermediate port for a specified number of days. The SAIL model can treat these stops in two ways. In one case the cargo remains on the same ship for both legs of its trip; in the other case, the cargo may travel on different ships for the two legs of its journey.

The planning analyst has control over these options. A single minimum layover time for changing ships may be specified for a planning scenario; that is, if a layover is less than the minimum, the cargo will not be allowed to change ships at the intermediate port. If the cargo has a layover time in excess of the minimum, it may reload onto a different ship for the continuation of its trip.

For cargos that do not change ships, the impact on the model is to insert another port into the ship itinerary, and to add the layover time to the ship's transit time. Cargos that may

change ships are more complicated. These cargos are, in effect, split into two cargos, one traveling from the port of embarkation to the intermediate port, and the other continuing to the port of debarkation. The cargo making the first leg of the trip is called the independent cargo; that making the second leg is the dependent cargo. Clearly, there is a sequence dependency between these two shipments; a pointer is established to link the two cargo records. Artificial arrival and reshipment dates are generated for the intermediate port. Independent cargos are aggregated only with other independent cargos; the same is true for dependent cargos.

Special checks are made to enforce the sequencing constraint of the independent and dependent legs in the Ship Routing Subsystem. When dependent cargo is a candidate for shipping, a search is made to ensure that an adequate amount of the associated independent cargo has already been shipped. However, due to timing considerations, some connections still may not be made.

The Simulation Subsystem must deal with these missed connections. For one thing, a parameter controls the number of days a ship carrying the second leg of a cargo will wait at an intermediate port for the first leg to arrive; if the delivery is not made in time, the ship leaves, stranding the cargo. Another parameter allows the user to specify that cargos that would be stranded in this manner should not leave their port of embarkation in the first place.

4.3.15 Attrition

The prospects for attrition can change over both time and space. Let us first consider geography; a system outside of the SAIL algorithm is first used to establish geographical regions where attrition may occur. A region is bounded by pairs of latitudes and longitudes; within the region a fraction is given which represents the probability that a ship in the area will be lost each day it is in the region. There is no limit to the number of such regions that can be defined and the regions may overlap. Rules must govern the overlaps; the earliest definition or the latest definition may apply, or the higher or lower attrition may be used.

The end result of the region definitions is a set of probabilities for each latitude and longitude throughout the world defining the risk of attrition for each one degree by one degree area. This information is contained in a binary file that will be read by SAIL during the distance computation module execution.

Changes of attrition risk through time is handled by creating a new file for each epoch. The number of changes will be limited by the dimensions of the program.

The risk a specific ship incurs while sailing a particular route is determined by the risk potentials through which it passes and the time of exposure. Hence, attrition is a function of ship speed. To accommodate this, a risk-versus-speed relationship is determined for each

port-to-port route. These functions can be approximated by a hyperbola with coefficients A and B:

$$risk\ probability = \frac{A}{speed^B}$$

The main computational process is, then, one of translating the risk map for a period into a set of A and B coefficients for each port-to-port route. This is done by dividing each such route into a series of short segments, determining the risk on each of these segments, and finally, accumulating these segment risks into an overall risk for the route. As the routes are used by ships, the individual ship speeds are used with the coefficients of the route to give the risk to the ship between the ports.

4.3.16 Ocean Avoidance

Ocean avoidance begins very much like attrition; the analyst, outside of the SAIL algorithm, defines "rectangular" areas through which ships are not to travel by pairs of latitudes and longitudes. There is no limit to the number of such areas and areas may overlap; virtually any shaped area can thus be constructed. (More free-form ways of defining avoidance or risk areas are not excluded; they are just harder to implement.) The ocean avoidance areas are transmitted to SAIL on the same file as the attrition risks; a special code is used to indicate that a latitude/longitude point is to be avoided. As with risks, ocean avoidance areas may change with time.

Implementation of avoidance areas in SAIL is accomplished in the Distance Module. For each time period defined, each link in the ocean network (see Paragraph 4.2.3) is checked to see if it passes through an avoidance area. If it does, the link is marked to be removed from the network. There is one exception: if a route cannot be completed without passing through an avoidance area, the route will be constructed with a minimum distance in the area. (This may occur, for example, if an avoidance area is defined which includes a port.)

4.3.17 Underutilized Ships

For each type of ship, the analyst may specify a minimum percentage of capacity which constitutes a viable load. Trips that will almost certainly fail to meet that standard are eliminated prior to setting up the loading problem in the Ship Loading Subsystem. A final removal of underutilized ships occurs following the detailed scheduling in the Simulation Subsystem--after intolerably late cargo is removed from the ships (see Paragraph 4.3.21).

4.3.18 Assault Follow-On Echelon (AFOE)

Cargo identified as AFOE cargo is not mixed on ships with non-AFOE cargo. Further, the ships must remain at the port of debarkation until a specified release date or for a specified number of days after delivering their cargo. (If both a release date and a number

of delay days are specified, the release date is used.) If the ships are not to be released, a large number should be specified for the release date.

4.3.19 At-Sea Unloading of Resupply Petroleum, or Consolidation of Load (CONSOL)

The unloading rate of resupply petroleum with at-sea ports of debarkation is half the normal rate, and delivery prior to a specified date is not permitted. The mechanisms are in place for implementing this feature, but a way of identifying a port as an at-sea location is needed in the port data.

4.3.20 Bunkering

Ships begin service with a specified fraction of a full load of fuel; fuel capacities and use rates are known for each ship; a minimum fraction of capacity before refueling is specified; and ports at which ships may bunker are given. If requested by the planner, a special Bunkering Module is called from the Simulation Subsystem which determines the ports on the ship's route at which bunkering must occur and the amount of fuel required at each. If bunker ports are not adequately spaced on the ship's route, diversions to bunker ports will be made. If a successful diversion is impossible, the ship is marked as stranded for lack of fuel.

The Reporting Subsystem accumulates the fuel used at each bunker port and reports the quantities by type and time period.

4.3.21 Unacceptably Late Cargo

A maximum number of days of lateness can be specified for each of several classifications of cargo: unit cargo, replacement personnel, resupply ammunition, resupply petroleum, and other resupply. Cargos delivered late by more than this maximum are deleted from the schedule and reported as shortfall. Determination of this critical lateness is made in three places. The Aggregation Subsystem may find that acceptable delivery is impossible under any circumstances. Second, in the Ship Routing Subsystem, cargos that clearly cannot be acceptably delivered are deleted; this provision allows the shipping resources to be allocated to other cargos. A final check is made in the Simulation Subsystem. After unacceptably late cargos are deleted, the simulation is repeated to compute new times for events and to display revised ship itineraries.

4.3.22 The Refine Module

As an option during the Simulation Subsystem, refinements to ship loading can be requested. This process looks at all late or undelivered cargo and attempts to find better opportunities on other ships. If a sufficiently attractive option is found for a cargo, the change is made. Experience indicates that this feature can dramatically improve the delivery of a few cargos, but has little effect on the overall plan statistics.

4.3.23 Minimum Port Activity

It is undesirable for a ship to stop at a port to load or unload trivial amounts of cargo; the analyst can define "trivial" for ports of embarkation and debarkation separately. Two actions are taken as a result of this feature. First, if a port has too little activity of all types within some reasonable time period, the cargos are marked as infeasible; that is, there is no way to load (or unload) those cargos within the defined minimum activity levels. Such cargos are removed from the schedule. Secondly, individual ships are not allowed to stop at a port for insufficient loading or unloading activity. This may result in cargo being late or stranded.

4.3.24 Special Treatment of Non-Combatant Evacuees (NEO)

NEO may be carried by any type of ship except tankers, but the passenger capacity of most cargo ships under emergency conditions is not well defined. A special stow factor for NEO is therefore given to permit these ships to carry reasonable numbers of evacuees.

It is generally not the intent, however, for cargo ships to be routed much out of their way to pick up evacuees. For this reason, the planner may specify a maximum distance a ship is allowed to go for this purpose. This constraint prevents cargo ships from making special trips for the sole purpose of loading NEO.

4.3.25 Lift Capability Computations

The SAIL algorithm was designed to analyze the ability of a known set of assets to transport a known set of movement requirements. If the assets are known but the requirements are not yet well-defined, a special lift capability analysis can be performed.

The analyst first defines channels, which in this case is defined as a path over which cargo moves, a class of cargo to be moved, and the percentage of total movement that the route and class should end up with. Geographic displays are particularly helpful in establishing these flow patterns and prototypes have been constructed.

The lift capability analysis then employs the available ships to move as much cargo of the appropriate classes as possible over the defined routes, maintaining the target percentages over each route. A cut-off date stops the process. Reports then give the amount of cargo that can be moved of each type over each route using the given assets.

4.3.26 Required Lift Analysis

Another special analysis can be used if the movement requirements are given but the lift assets have not been established. In this case, a file of nominal ships is first constructed. These are "pure" ships; that is, they have only one kind of compartment: all breakbulk, all container, all RO/RO, etc. This file of nominal ships is constructed so that plenty of ships of each type is available.

In addition to having adequate ships, the special code associated with this feature places no constraint on the ship availability date. At this point, normal routing and loading takes over. The result is a report of the number of each type of nominal ship required, where

each ship is first needed, and the resulting delivery profile. Ships are re-used whenever possible to accomplish on-time delivery. It should be noted that, even with the lift assets so unconstrained, on-time delivery is not guaranteed; late available to load dates, unrealistic delivery dates, port throughput constraints, or the lack of proper port facilities can still result in scheduling problems.

There are at least two uses of this report. First, the analyst may find it useful in establishing the requirement for real ships by matching nominal ships used to real ships available. Secondly, the results will display scheduling deficiencies unrelated to ship availability, allowing other problems to be isolated and identified.

4.3.27 Multiple Plan Capability

An analyst may wish to assess the impact of simultaneous execution of several operation plans. The SAIL algorithm will accept requirements (cargo) data for several plans; some or all of the ships may be dedicated to one or more particular plans in the set. Each plan may begin at a different time relative to the start date for the entire suite of plans. For example, a defensive naval operation may begin at one time with certain lift resources dedicated to that action. At a later date, a plan for an offensive operation may be initiated, using other specific ships. There may be other ships available to either phase of the operation.

4.3.28 Special Scheduler for Sparse Shipping

Underutilized ships (Paragraph 4.3.17) and unacceptably late cargo (Paragraph 4.3.21) may result in a voyage being removed from the schedule relatively late in the scheduling process. The ship thus released from service may end up idle while other ships are making multiple voyages that deliver cargo later than desired. To overcome this problem, a special sequencing of the operations of the SAIL algorithm may be invoked which gives the idle ships an opportunity for gainful re-employment. This feature is not recommended for routine use since it is more time consuming and shows no improvement except where ships are very scarce.

4.3.29 The Transfer Port Concept

The analyst may wish to restrict certain ships from entering particular ports; for example, some ships may be available on the understanding that they not be used too close to hostile activity. These ships may carry materiel to a nearby transfer port; other ships will move the cargos on to their original ports of debarkation. By using a special version of SAIL, this novel concept of operations can be modelled. The analyst specifies which ships are restricted, which port is the transfer port, and which ports are off limits to the specified ships. The algorithm tries to find cargos for the restricted ships for which there is adequate lead-time to permit the additional handling time at the transfer port, and (more importantly) accommodate the more complex ship scheduling problem generated by the transfer operation.

4.3.30 Experimental Scheduling from CONUS Complexes

An experimental method for aggregating cargos through CONUS complexes can now be tested. Three complexes, East Coast, Gulf Coast, and West Coast, are described in the first three records of the port descriptor file (PORTDAT). The port identifier of the first record should be a "2", which is the first character of the complex code for all ports on the East Coast. (This special identifier is used instead of the GEOLOC code normally found in this field.) The second and third ports' names are "3" and "5", which correspond to the Gulf Coast and West Coast respectively. The addition of these three records is all that needs to be done to invoke the experiment.

All cargos with POEs using these complexes will automatically be routed through the complex instead of through the individual ports. In this experimental implementation the identifiers must be defined exactly as described; if they are not, no CONUS complex aggregation will take place. If desired, a more flexible scheme can be developed that will replace any set of ports with their common complex.

4.4 SOME COMMENTS ON COST FUNCTIONS

References have already been made to the cost functions used in the evaluation processes of the Ship Routing and Ship Loading Subsystems. These functions essentially implement the objective of the optimization problems. Experimentation with objectives is expected to continue, so these functions are not necessarily fixed; in this paragraph the way in which they are currently formed will be indicated.

Objectives are subjective! The kind of answer the optimization problem will seek is a reflection of the value system of the analyst that formed the problem. We have already discussed that, in the sealift problem, there are multiple, competing objectives: avoid lateness, avoid earliness, use proper stowage compartments, use few ships, maintain unit integrity, etc.

Within each of these sub-objectives it is easy to rank the options; for example, delivering a cargo 5 days late is always worse than delivering it 3 days late, which is always worse than one day late. It is not so easy, however, to say *how much worse* 5 days late is than 3 days late, or how that difference compares to the 3-day to 1-day increment.

The situation becomes more complicated when two sub-objectives are involved. Which is better: to use two partly-loaded ships that get all the cargo to its destinations on time, or use one fully-loaded ship with the result that "some" of the cargo is "a little" late? How does one define "some" and "a little?"

In the current software, these value judgements and trade-offs are not adequately reconciled. Our approach has been to form objectives that are relatively simple, unambiguous, and (hopefully) predictable in their impact on the solution. The approach is the following:

1. The sub-objectives are identified; within each, two or more levels of performance are specified. For example, one sub-objective is lateness; the levels might be: more than 10 days late, 5 to 10 days late, less than 5 days late, and not late.

2. The sub-objectives are ranked according to their importance. Avoiding lateness is ranked first; in the current scheme, this objective will not be compromised to achieve any other sub-objective. This is the case all the way down the rankings; no higher level sub-objective will be jeopardized in favor of a lower-level goal.

3. All combinations of the levels of sub-objectives are formed and a cost associated with each. From (1) and (2), an ordering of these combinations is implied; that is, we can say that one combination is better than another. But we still cannot say how much better. For now, we simply establish the order of the combinations, assign a cost of zero to the most favorable, a cost of one to the next most favorable, and so on.

The way this approach is now used to form cost functions for the Ship Routing and Ship Loading Subsystems are shown in Fig. 5.3 (discussed in Paragraph 5.1.6.5 and Section 5.5). One final note: we use the word "cost" rather loosely in this context; we are actually forming penalty functions. The values we arrive at bear no relation to any real-world costs. Clearly, many of the objectives do not lend themselves to such a financial measure. There is no way to put an economic price on the late delivery of a critical military unit, for example. What we are trying to do is provide a means of evaluating the efficacy of various potential solutions to a problem in the face of competing objectives.

5. THE SAIL MODEL - IMPLEMENTATION

This section deals with the coding, computing resources, parameters, and files involved in the implementation of the SAIL sealift scheduling model. As testing continues, particularly on operational data, changes to the software are expected to be required. A few additional features and capabilities may also be added. At this stage of development, however, the rate of modification is expected to taper off; major algorithmic approaches and software organization are reasonably well established.

This section begins with a description of the subroutines involved, grouped by major subsystems. The input and output files are then given in detail, followed by the parameters and options under user control. The computer resources required to run the model are discussed, along with the impact of problem size on the dimensions of the model's vectors and arrays.

5.1 THE SAIL SOFTWARE

The SAIL system is coded in FORTRAN 77. A main program controls the overall flow through the model; there are blocks of subroutines that implement each of the several subsystems discussed in Section 4. The code for each of these subsystems will be discussed in the following paragraphs.

For convenience, common blocks are maintained separately from the executable code; programs use INCLUDE statements to insert the proper commons where they are needed. On some systems partitioned data sets are necessary for proper inclusion. The commons contain descriptions of the variables used and notes concerning the dimensioning of those variables; the centralization of all variables should facilitate redimensioning of the problem.

The bulk of the computations use integer variables, so most routines begin with an IMPLICIT INTEGER (A-Z) statement.

5.1.1 The MAIN Program

Overall control of the execution of SAIL is provided by the MAIN program. Variables and files are initialized, and calls are made to invoke the major subsystems. The following is an ordered list of the subroutines called by MAIN and a very brief description of each. More extensive treatments of each subsystem appear in later paragraphs.

Subroutine INIDIM is called first to initialize the variables that define the current dimensions of the SAIL model. This initial step is followed by a series of calls to routines that read the data prepared for SAIL by the data management system in SEASTRAT; these input routines constitute the Interface Subsystem.

Subroutine DISCAL, which is the driving routine for the Distance Computation Subsystem, is called to calculate the interport distances and the initial ship availability dates. When repeating experiments using the same ports, the distance file can be saved and reused to avoid recalculating the distances. Also, if the same ship file is used, the ship availability dates can be saved and reused. (See Paragraphs 5.1.4.11 and 5.1.4.12.)

The next several calls exercise the Aggregation Subsystem. Subroutine ELIM will eliminate cargos that are irretrievably late and write a record for each on the EXCLOUT file. Calls to other subroutines set up the time intervals for the run (INTERV), compute nominal port queue times (PODQ and POEQ), and set up the two legs of the journey for cargos with intermediate stops (SPLIT). Key to this subsystem, though, is the call to subroutine CHANEL, which controls the aggregation process.

The Ship Routing and Ship Loading Subsystems are invoked by a call to Subroutine ROUTCON, which is explained in more detail below. Calls to SIMOVR and POST then execute the Simulation and Reporting Subsystems respectively.

5.1.1.1 Subroutine INIDIM

This routine defines the dimension size of all arrays in common blocks. These variables are used throughout SAIL to assure array limits are not exceeded.

5.1.1.2 Subroutine ROUTCON

This routine controls the route development process. It checks to see if dry cargo and petroleum problems exist and invokes the routing algorithm for those that do. The parameter SWNOPT is checked to see if heuristic or optimal-based routing procedures are to be used.

5.1.1.3 Subroutine REDSOL

This routine reads a ship routing solution from the file specified by the variable SOLGET.

5.1.1.4 Subroutine WRTSOL

This routine writes a ship routing solution to the file specified by the variable SOLSAV.

5.1.1.5 Logical Function INFEAS

This functions checks to see if a particular channel can be loaded on a given ship compartment at a given port facility. If it cannot, the function is returned TRUE (that is, the combination is infeasible).

5.1.1.6 Subroutine SOLU

When called, this routine writes the current solution vectors for all problem types on line. This is used for debugging or special problem analysis only; the routine should be called with discretion in that very large amounts of output can be generated.

5.1.1.7 Subroutine DELETE

This routine deletes ships from the solution vector that fail to have an acceptable load (as defined in Subroutine ADEQAT); these solution vector entries are deleted prior to setting up the ship loading problem.

5.1.1.8 Subroutine NITIAL

Several arrays are initialized for the input programs.

5.1.1.9 Subroutine INTHRU

The changes to port throughput during the plan are contained on file THRUDAT. This routine reads the THRUDAT file and sets the time-varying representation of port throughput capacity to the proper values for each facility type.

5.1.1.10 Integer Function ANYCAP

This function checks to see if a port facility has any capacity at any time during the schedule. It is used to determine infeasible load or unload situations.

5.1.1.11 Subroutine PORCHK

This routine tests whether or not appropriate port facilities exist for each cargo; if no facilities are found, the cargo is eliminated from the scheduling process and a message is written to the exception file (EXEPOUT).

5.1.2 The Interface Subsystem

A separate subroutine is used to read each type of data. The various input routines, the files they read, and a brief description of the information on the files are given in Table 5.1. More information about each file, including the record structures, is given in Paragraph 5.2. Limited data editing and consistency checks are made as information is read and stored into FORTRAN variables.

5.1.3 The Aggregation Subsystem Routines

5.1.3.1 Subroutine SPLIT

The cargo list is searched for movements that have intermediate stops of sufficient duration to permit the consideration of using different ships for the two legs of the trip (see parameter, IDMIN, Paragraph 5.3.5.4). These cargos are split into two cargos for scheduling

purposes. The first cargo, which retains the original cargo record identification number, is scheduled for delivery at the intermediate port. A latest delivery date is created which permits the cargo to be held in the port for the required number of days and then be reloaded and shipped to its final destination. If there is not enough time for all of these events to occur prior to the required delivery date at the original port of debarkation (POD), a compromise time is found and a message is written on the EXEPOUT file.

A new cargo is then created for the second leg of the trip. Its port of embarkation is the intermediate port, and its port of debarkation is the original POD. A ready-to-load date is computed, but the original latest arrival date at the POD is used. This "created" cargo has all the same attributes as the original cargo. It is given the next available sequential cargo index number. Pointers between the two legs are established, permitting other routines to recognize the dependency between the two cargos.

5.1.3.2 Subroutine ELIM

This routine computes the earliest arrival date of each cargo at the port of debarkation, assuming the best available ship is used solely for this cargo. Cargos that cannot be delivered within the prescribed maximum lateness for their classes are eliminated from the problem and reported as such on the exclusion file (EXCLOUT).

Table 5.1. Input Program Descriptions

Subroutine	File Read	Contents
INPARA	PARADAT	Scheduling policy and modeling parameters, canal dimensions, and default values
	PLANDAT	Parameters controlled by the analyst
INSHIP	SHIPDAT	Descriptions of ships that can be used in the plan under analysis
INPORT	PORTDAT	Description of ports in the plan
INCGO	CRGODAT	Description of movement requirements
INSTOW	LOADDAT	Load rates of cargo onto ships, usually in measurement tons per day, by ship type and cargo class
	UNLDDAT	Unload rates
	STOWDAT	Stow factors, in hundredths, by ship type and cargo class
	FACIDAT	Type of port facility needed to load or unload a class of cargo on a particular ship compartment

5.1.3.3 Subroutine INTERV

Time is divided into intervals for cargo aggregation and port queuing estimations. It may be desirable to have more precise estimates early in the schedule and more relaxed estimates later. (This flexibility allows a greater degree of cargo aggregation in later intervals.) Three parameters are used to create the intervals: first is the interval width, in days, at the beginning of the run; second, the crossover day between the first and last interval widths; and finally, the interval width in the later part of the run.

5.1.3.4 Subroutine PODQ

During the aggregation and routing processes, estimates of queuing delays at ports are needed for each time interval. This routine estimates the delays at ports of debarkation.

The demand placed on the throughput of each facility is estimated for each time period; the assumption is made that materiel will arrive at the port on its latest arrival date and that the type of port facility used (general, container, etc.) will be that most favored by the cargo class involved. In periods where the demand exceeds the capacity, a queuing delay is expected, the length of which is proportional to the excess of demand over capacity. Delays may accumulate from one period to the next if there is a continuing pattern of high demand.

5.1.3.5 Subroutine POEQ

This routine is similar to PODQ, except queuing delays at ports of embarkation are estimated. This is done after the channels are formed and nominal sailing dates are established. These dates are used to estimate the demand on throughput at all facilities. In the event that a port has both loading and unloading functions, the sum of the demand is used to estimate the queuing.

5.1.3.6 Subroutine CHANEL

CHANEL aggregates cargos with similar characteristics into entities called channels. During the ship routing phase, individual cargos lose their identities while channels are assigned to ships. The main purpose of this step is to reduce the size of the problem for the optimization routines and to enhance the likelihood that related cargos will be loaded together.

All cargos in a channel must have

1. the same ports of embarkation;
2. the same ports of debarkation;
3. the same departure interval;
4. unit discrimination (the analyst has three choices): a) aggregate only cargos with the same unit description, b) aggregate on the basis of unit or non-unit cargo, or c) ignore unit descriptions;
5. the same cargo class;
6. similar available to load dates;
7. the same intermediate ports, if any;

8. must be either AFOE or non-AFOE cargo; and,
9. must be either CONSOL or non-CONSOL cargo.

It is also useful to aggregate channels with common traits into groups called clusters. All members of a cluster share

1. nearby ports of embarkation;
2. nearby ports of debarkation;
3. the same intermediate port, if any;
4. the same shipping interval;
5. and are either all ammunition, or all non- ammunition;
6. are all AFOE or all non-AFOE cargo; and,
7. are all CONSOL or all non-CONSOL cargo.

5.1.3.7 Subroutine WREXCL

This routine records the cargos that will not be delivered on file EXCLOUT. See Paragraph 5.6 for a description of the exclusion codes.

5.1.3.8 Subroutine FINDIN

FINDIN accepts an integer time and returns the interval number in which that time falls.

5.1.3.9 Subroutine INITCH

Channel vectors are initialized, and the number of channels is set to zero.

5.1.3.10 Subroutine CLUSTR

Clusters are aggregations of channels that involve nearby ports, are either ammo or non-ammo, CONSOL or non-CONSOL, AFOE or non-AFOE, on the same preferred ship (if any), and the time intervals must be close enough together to overlap. This routine finds and marks those collections of channels.

5.1.3.11 Subroutine PCLUS

At the analyst's discretion, ships will not make a port stop for less than a specified amount of cargo. This routine aggregates channels at ports for the purpose of determining if adequate amounts are to be moved. This is done by looking at the maximum amount of cargo that could be placed on a single ship (given cargo classes and other restrictions). If this amount is insufficient, a ship will not stop at the port and the cargo cannot be delivered.

5.1.3.12 Subroutine CULL

This routine evaluates the aggregations created in subroutine PCLUS. If an aggregation is less than the required minimum quantity for a ship to stop, the channels

composing the aggregation are removed from the problem and the cargos that make up those channels are marked as undeliverable.

5.1.4 The Distance Computation Subsystem

5.1.4.1 Subroutine DISCAL

This routine is the control program for the interport distance calculations. INNODE, INLINK, and INLINE are called to read the global network data. ADPORT merges the port locations into the grid system. Then STAR and DISFIL are called for each of the possible canal status combinations producing the four interport distance tables: (1) for both canals open; (2) Panama closed; (3) Suez closed; and (4) both closed. A canal is closed in the shortest path algorithm by assigning the canal link a distance of 50,000 nautical miles, thereby denying it a position on any shortest path.

5.1.4.2 Subroutine SADS

SADS calculates the initial availability of each ship at every port. Appendix H describes the rules for this computation. Basically, peace or war conditions and fleet designations dictate early warning days and administrative delays, and also whether a ship will continue to its initial destination or be immediately diverted.

5.1.4.3 Subroutine DISCHK

DISCHK computes the distances from any node to all other nodes, taking into consideration the starting time and whether or not the canals are open.

5.1.4.4 Subroutine CANFND

To evaluate the effect of canal status on transit times, CANFND determines the order in which canals are traversed.

5.1.4.5 Integer Function DIST

Given a departure port, a destination port, and the appropriate table number (which indicates the canal status) DIST will return the distance between ports and which canal, if any, will be traversed.

5.1.4.6 Subroutine SPA

SPA is a branch-and-bound shortest-path algorithm used to calculate point-to-point distances through the global network.

5.1.4.7 Subroutine INLINK

LINKDAT, the file describing the node connections in the distance network, is read, and link distances are calculated, if necessary. Note that LINK(1) is reserved for the Panama

Canal and LINK(2) for the Suez Canal so that canals can readily be opened and closed by altering the distance on these links.

5.1.4.8 Integer Function GCD

GCD will return the great circle distance, in nautical miles, between any two global locations described in the argument list by their latitudes and longitudes.

5.1.4.9 Subroutine NEAR

NEAR will return the network node closest to any given global location.

5.1.4.10 Subroutine STAR

STAR builds the indexes for each node showing all other nodes linked to it. These are called the "forward star tables" in the shortest-path algorithm.

5.1.4.11 Subroutine SAVDIS

The distances and initial ship availabilities may be saved on a file specified in PARADAT.

5.1.4.12 Subroutine GETDIS

When successive SAIL runs are made using the same ports and initial ship locations, the distance calculations need not be repeated. On any run, a file name may be specified in the PARADAT file for saving the distances and the ship availability dates. Subsequent runs could specify in PARADAT that this distance file be used in preference to recalculation. GETDIS is the routine that will read the specified distance file.

5.1.4.13 Subroutine ADPORT

ADPORT incorporates the port locations into the node definitions. Each port location is compared with the node locations; if a node is found to be close enough to the port location, as defined in variable DELTAD, no new node or link will be added. But if the port location is not found in the node list, a new node will be appended and a new grid link will be added that will connect the new node to its nearest network neighbor. After ADPORT executes, all ports will be included in the world map distance network.

5.1.4.14 Subroutine INNODE

This routine reads file NODEDAT to obtain all the global locations, or nodes, to be included in the network.

5.1.4.15 Subroutine DISFIL

As described in Paragraph 5.1.4.1, there are four interport distance tables, one for each combination of canal status. DISFIL accepts the desired table number and builds the appropriate interport distance table.

5.1.4.16 Logical Function NTRSEX

This routine determines whether travel between two global locations, defined by latitude-longitude pairs in the argument list, would cross any of the lines defined in the LINEDAT file. Three assumptions are currently made: the two points are no more than 30 degrees apart; the lines are no more than 40 degrees long; and the lines do not cross longitude 180.

5.1.4.17 Subroutine INLINE

File LINEDAT, containing the global lines that cannot be crossed, is read by this routine.

5.1.4.18 Subroutine DISTIM

This important subroutine is called from many points in SAIL whenever distances between ports and sailing times are needed. Given a starting port, the departure time, a destination, and a ship, DISTIM returns the distance traveled and the expected arrival date at the destination, accounting for sailing time at the ship's nominal speed, canal status, and canal transit time.

5.1.4.19 Logical Function COPEN

This logical function accepts the expected C-day of arrival at a canal and returns the canal status for that day (TRUE, if open).

5.1.4.20 Subroutine TABLED

When the interport distances have been calculated for all canal combinations, TABLED is called to save them on file INTPORT.

5.1.4.21 Subroutine CROUTE

This routine finds the canals (if any) used on a given route.

5.1.4.22 Subroutine SAD1

Ship availability dates are continually updated throughout the SAIL procedures. The initial ship availability dates are stored by this program for future use.

5.1.4.23 Subroutine RISKS

This program determines the risk probabilities between any two positions on the globe. It first finds points along a great circle route between the positions spaced no more than a specified number of miles apart (100 miles, for example). Risks are assessed at each point using data from the risk data file and an overall risk function for the route between the positions is calculated. The actual risk will be a function of the ship's speed, so the routine computes coefficients a and b which describe the relationship between speed and risk for a sailing between the given positions:

$$Risk = \frac{a}{speed^b}$$

5.1.4.24 Subroutine DECOMP

Attrition risks are described as the probability, P, that a ship will be lost each day that it is in a risk area. This routine computes the probability, X, that a ship will be lost while travelling at a known speed, S, between two points in the risk region which are D miles apart. This is accomplished by solving the following equation for X:

$$1.- (1-X)^N = P$$

where N is the number of times the distance D can be traversed in a day. Newton's method is used.

5.1.4.25 Subroutine HYPERB

The relationship between ship speed and risk is represented as a hyperbola. This routine finds the coefficients A and B of a hyperbola that goes through two points, (x1,y1) and (x2,y2). The form of the equation is

$$y = \frac{A}{x^B}$$

5.1.4.26 Subroutine INIRSK

This routine initializes the A-coefficient of risk for each link in the ocean network to -1; this indicates that the risk for the link has not been computed. The risk grid for the time phase being considered is then read from the appropriate risk data file.

5.1.4.27 Subroutine LNKRSK

This routine finds the latitudes and longitudes of the two ends of an ocean network link and calls subroutine RISKIT to find the coefficients of risk for the link.

5.1.4.28 Subroutine RISKIT

The risk coefficients (relating risk to ship speed) for individual ocean network links are found by this routine.

5.1.4.29 Subroutine PTOP

This routine finds the distance and risk coefficients between two ports during a time phase (that is, a set of canal conditions and a specified risk definition file).

5.1.4.30 Subroutine TOPOES

In the process of finding initial ship availability dates, a ship's position prior to being available for use is first determined. Then, this routine finds the time required for the ship to get from that position to each port in the problem.

5.1.4.31 Subroutine CASE1

This is a special computation for ship availability dates for peacetime conditions when the ship is in non-OPLAN use at the beginning of the plan. See Appendix H for details.

5.1.4.32 Subroutine SADTAB

This routine builds a file, SADSOUT, of ship availability dates and the distances between each ship's initial position and the various ports. The routine is called only when stand-alone distance and ship availability date computations are called for.

5.1.4.33 Subroutine INIDIS

Initializations of several arrays are performed by this program, including the risk arrays, the distance array, and a vector that is used as a pointer into these arrays. This pointer is explained in the comments in COMMON DISCOM.

5.1.4.34 Subroutine ARC

This routine finds N equally spaced points along a great circle route between two points. The number of points is such that the maximum distance between them does not exceed a specified number of miles.

5.1.4.35 Subroutine AVOID

This program checks to see if there are any ocean avoidance areas and, if there are, modifies the ocean network accordingly. Ocean avoidance is achieved by setting mile

multipliers for each link which traverses an avoidance area to a large number. This is then used in the shortest path algorithm to ensure the link will not be used.

5.1.4.36 Subroutine INVIEW

The VIEWDAT file is read by this routine. The first record in this file describes an initial set of canal conditions and the name of the risk and ocean avoidance file to be used at the beginning of the plan. Subsequent records give dates and conditions when changes occur. Each of these sets of changes defines a time phase for the plan.

5.1.4.37 Real Function UNTIME

This function finds a nominal unload time for a given ship, based on directions given by MSC (memo from J. K. Wiswesser, 17 May 1990).

5.1.4.38 Subroutine FNTIME

For any C-day, this routine finds the time phase in which that day falls.

5.1.4.39 Integer Function DISBAS

Given two ports and the time phase of departure from the first port, this function returns the distance traveled and the canals used. The name stands for "distance basic;" the routine differs from DISTIM in that it can be used when the specific ship (and hence the speed of travel) is not known.

5.1.4.40 Real Function RISKY

This function retrieves the attrition risk based on a port-to-port route, the time of departure, and the speed of the ship.

5.1.4.41 Real Function RISTIM

Given departure and arrival ports, a time of departure, and a specific ship, this function computes the probability that the ship will suffer attrition. No provision is made for the risk changing during the journey. This process uses the function RISKY for making the risk determination.

5.1.4.42 Subroutine DISCPY

Paragraph 5.1.4.36 dealt with the routine that reads in the different canal conditions, risk areas, and ocean avoidance areas. When two time periods differ only in canal conditions or risks, there is no difference in the basic distances (port-to-port distances not using canals plus the port-to-canal distances); subroutine DISCPY identifies cases where a time phase will have identical distances to the previous phase and copies the distances rather than recomputing them.

5.1.4.43 Subroutine SNEAR

This routine is used to connect a ship in an arbitrary ocean location onto the ocean network. Unlike subroutine NEAR, which would connect the ship to the closest node, subroutine SNEAR finds a close node in the direction the ship needs to travel to reach its current destination. There is a trade-off between the nearness of the nodes and the deviation from the desired line of travel; to assure consistently feasible connections (ones that do not cross land), somewhat more emphasis is placed on the closeness of nodes.

5.1.4.44 Subroutine COMNOD

This routine is used by subroutine SNEAR to determine if a candidate node for a ship to connect to is better than the best node found so far.

5.1.4.45 Subroutine SADDUN

The computation of ship availability dates is a time-consuming process. When this process is initiated for a ship, subroutine SADDUN checks to see if a similar ship has already had availability dates computed. If so, the previous results are used. Ships must be similar in all respects for this efficiency to apply.

5.1.4.46 Subroutine KEEPRT

This routine builds an array of latitudes and longitudes representing the routes between ports. These routes can be written to a file which can be accessed by a plotting program in order to show port-to-port routes or the path taken by a particular ship. (For more information, see the route and pointer file descriptions in Section 5.2.)

5.1.4.47 Subroutine SHPRTE

This routine is similar to the preceding; it finds the latitude and longitude positions of a ship as it moves from its initial location to its initial destination or first port of embarkation.

5.1.4.48 Subroutine INWORL

The *FAST* system provides the capability to describe ocean avoidance or risk areas using interactive geographic screens. The output of that system is a file called (as of this writing) WORLD.ATT. If a file by that name exists, SAIL, using Subroutine INWORL, will read it instead of the RISKDAT file to determine avoidance and risk areas.

5.1.5 The Ship Routing Subsystem

5.1.5.1 Subroutine ACYCLE

This is the control routine for the optimization-based ship routing process. It calls MATGEN to set up the transportation problem and FIRSTX and SOLVE to obtain an initial solution. It then makes repeated calls to APRIME to find loads for one ship at a time; after

each ship is loaded and routed, subroutine ASAILM is called to modify the cost matrix and resolve the optimization problem. Subroutine DONE is called to see if the routing is complete; if not, the process is repeated. When finished, permanent solution vectors are constructed by subroutine BLDSOL.

All programs in the routing process honor the special ammunition handling rules: mixing or nonmixing of ammunition on ships is an option; if mixed, ammunition will be the last cargo loaded and the first cargo unloaded; breakbulk ammunition will be carried on sheathed ships only; all ships are initially considered unsheathed, and a delay will be imposed for sheathing.

5.1.5.2 Subroutine APRIME

This program examines the transportation problem solution, looking for a good candidate channel-to-ship assignment. Decision rules are used to find the best such candidate, called the prime channel. Best is defined in terms of both quantity of cargo and urgency of shipping dates. Subroutine CHINIT is called to ensure that the prime cargo will be delivered within the lateness criterion and to initialize the ship's route. An unsuitable prime will be rejected at this point, and the process of finding a new prime would begin again.

If the ship is not filled at this point, subroutine SECON1 is called to find channels that are geographically incompatible with the prime; all such channels have their costs set to infinity for the ship being routed and the transportation problem is re-optimized. Secondary channels are then selected from the solution vector in the order that minimizes the number of port stops for the ship; each candidate is tested by subroutine CHEVAL to see if it will be delivered on time and to see what impact the addition of the new channel will have on the channels already on the ship.

Figure 5.1 shows the function that is used in this latter evaluation. Whether the candidate is added to the route or not depends on three things: the amount of the candidate compared with the amount of cargo already on board, the impact of the candidate on the timely delivery of the cargo on board, and the urgency of adding the candidate to this particular ship. If the urgency of adding the candidate is low, the decision will be made "cautiously," if the urgency is high, it falls into the "generous" category. In between, a "discriminating" choice is made. The slopes of the lines depicted in Figure 5.1 are under parameter control.

Candidates are added (or rejected) until the candidates in the solution vector are exhausted or until the ship is full; control is then passed back to subroutine ACYCLE.

5.1.5.3 Subroutine ASAILM

After APRIME has assembled the ship route, ASAILM computes new ship availability dates for the sailed ship, calls routines to restructure the transportation problem for the next iteration, does some bookkeeping work on permanent solution vectors, resets some variables, and updates the trip counter for the ship.

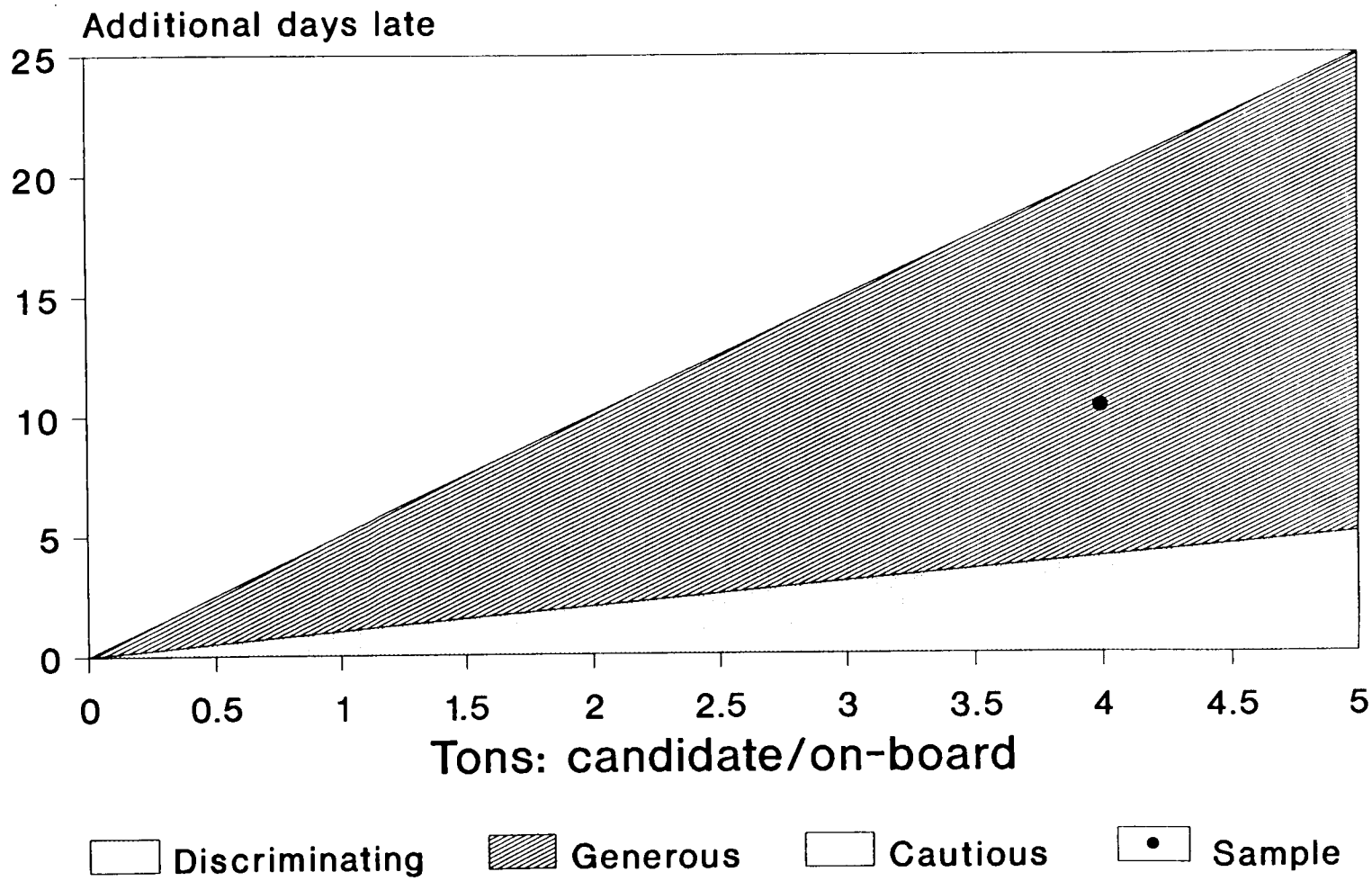


Fig. 5.1. Acceptance criteria for secondary channels.

5.1.5.4 Subroutine NCYCLE

NCYCLE is the control routine, similar to ACYCLE, for the heuristic ship routing system.

5.1.5.5 Subroutine NPRIME

This routine puts together the route for a single ship trip in the heuristic routing option. It begins by selecting the next most critical, or prime, channel which is selected based on the remaining unshipped amount of cargo and the required shipping time. The lowest cost ship for that channel is then selected, and other ships with costs nearly as low are also marked. The prime channel and all channels clustered with it are trial loaded on these ships. The ship that most effectively carries these channels with the least remaining space (if any) is selected for sailing. If the ship is not filled by the clustered channels, other compatible channels are selected using subroutine NXTBST and tested using CHEVAL.

5.1.5.6 Subroutine NSAILM

When a complete trip has been put together by subroutine NPRIME, NSAILM adds the current trip assignments to the solution vector, computes the times the ship is available at all ports for its next trip, and prepares some arrays for the next iteration.

5.1.5.7 Logical Function FULL

This function checks to see if a ship is filled; if so, it is returned TRUE. It is used only in the heuristic routing option.

5.1.5.8 Subroutine STASH

This routine is used in the heuristic routing to build a temporary solution vector representing the storage of a particular channel in a ship compartment. These temporary vectors are accumulated into a permanent solution by subroutine NSAILM.

5.1.5.9 Subroutine NXTBST

This routine looks for the next best channel to put on a ship during the heuristic routing process. The first criteria are to choose channels that use ports the ship already visits, or if none, choose channels that use ports close to those already visited. Lower-level criteria are to select channels with earlier sailing dates, compatible available to load dates, and preferred assignments on the ship.

5.1.5.10 Subroutine EVLATE

Subroutine CHEVAL determines the lateness of a channel that is a candidate for being added to a ship trip. EVLATE evaluates that lateness and the alternative ways the channel can be carried to see if it should be added. This is not a final decision; the channel may still be rejected because of its impact on channels already on the ship.

5.1.5.11 Subroutine TEMPLD

This routine is used in the trial loading of channels. It loads a given amount of cargo into a compartment using the proper stow factor, incrementing the amount of the compartment's space that is used, and decrementing the amount of the channel remaining.

5.1.5.12 Subroutine PURGE

After a ship is routed by subroutine APRIME, the quantities of each channel placed on that ship are deleted from the transportation problem. This routine removes those amounts from both the demand vector and the solution vector. This is accomplished by driving a sufficient amount of the cargo into the dummy ship row; the supply of this row is also decremented, keeping the transportation problem in balance without altering real ship supplies.

5.1.5.13 Subroutine COLSOL

This routine finds the indices of the solution vector in a particular column of the transportation problem.

5.1.5.14 Subroutine MATGEN

This is the program that sets up the transportation problem representation of the ship routing subsystem. We think of this problem in matrix form; the columns of the matrix represent the channels of materiel to be carried, and the rows represent compartments of ships. There is a variable of the underlying linear programming problem at each row-column intersection. The characteristics of the problem are the following:

1. Each row (compartment) has a capacity, called the supply.

2. Each column (channel) has a quantity, called the demand.

3. A value at the intersection of a row and column (called an element of the matrix) represents the amount of the column's channel carried by the row's ship compartment.

4. There is a dummy row with a large supply which represents uncarried demand.

5. There is a dummy column with large demand which represents unused space on ships.

6. In the dry cargo problem, all quantities are in measurement tons (or passengers).

7. In the petroleum problem, all quantities are in hundreds of barrels (CBBLS).

8. In the transportation problem, the sum of all supplies must equal the sum of all demands.

9. The number of elements in the solution will be exactly the number of rows plus the number of columns minus one.

Throughout this document the term "solution vector" is referenced. By a solution vector, or a solution to the optimization problem, we mean the following: there is a set of three vectors whose corresponding elements represent a quantity assigned to a ship, a row representing the ship compartment, and a column representing the channel assigned. These three vectors are, respectively, X, PR, and PC. (These vectors are further defined in the Common Variable Definitions in Appendix F.)

Subroutine MATGEN sets up separate problems for dry cargo and petroleum. In either case, it looks through the list of channels, picking out only those appropriate to the problem being set up, and forms columns. It then chooses the proper ships from the ship file and forms rows from the ship compartments. A dummy row and column are also formed.

Costs are then assigned to all combinations of rows and columns. If a channel cannot be placed on a ship for any reason, the cost for the corresponding row-column intersection is set to a very high value. Reasons for infeasibility include illegal cargo class-to-compartment-type combinations, ships that cannot enter the ports through which the cargo must pass, and combinations of cargo and ships for which proper port facilities are lacking. For all legitimate combinations, the function TPCOST is called to find the cost.

When the optimization-based routing is used, the costs of all legitimate channel-ship combinations are increased by a fixed amount, called the cost plateau. When a ship is chosen for routing, this cost will be removed for only the chosen ship, thus giving that ship an advantage over all other ships. The result is that the ship selected for routing will have an improved opportunity to be filled to capacity.

5.1.5.15 Subroutine MATREV

This routine is used to modify costs in the transportation problem between successive solutions as the routing proceeds. Arguments to the subroutine indicate the conditions under which the recomputation occurs: whether or not to add the cost plateau to the costs, whether or not to check for the legitimacy of carrying evacuees, and the type of routing algorithm in use. The costs for only one ship are modified by each call to MATREV.

5.1.5.16 Subroutine TPCOST

Each cell in the transportation problem has a cost associated with it that is calculated by the Function TPCOST (see Paragraph 5.6 for the current parameters used in this process). This cost function is hierarchical in nature, that is, a series of levels have been established, each of which dominate all lower levels. The levels thus established, in order of importance are: lateness, earliness, ship utilization, compartment compatibility, ship proximity to port, cargo quantity to ship capacity ratio, and finally, ship sheathing for ammunition. The specifics described in Table 5.11 are expected to evolve as more experience is acquired but the hierarchical approach seems to be effective.

5.1.5.17 Subroutine COLPTR

This routine builds pointers to the active columns in each row of the transportation problem, thus permitting more efficient searches for entering variables in the solution algorithm. An active column is one in which the costs are in the feasible range and there is a non-zero demand remaining.

5.1.5.18 Subroutine FIRSTX

The solution procedure for the transportation problem requires that an initial feasible solution exists. Feasible means that the sum of solution quantities in each row and column equal the supplies or demands respectively; also, the solution vector must be the correct length; that is, there must be the correct number of elements of the matrix with quantities in them. FIRSTX obtains this initial solution.

The solution will not be optimal, but the process tries to find a good solution. The approach is to put quantities into the lowest cost elements first, building a feasible solution using progressively higher cost elements.

5.1.5.19 Subroutine SOLVE

Subroutine SOLVE implements a tailored version of the transportation problem algorithm. The algorithm works by starting with an initial solution. It then looks for opportunities to improve the solution by bringing a new variable into the solution (and getting rid of an old one). It is the search for new entering variables that is, by far, the most time-consuming part of the method. To speed up the solution process, SOLVE uses information known about the routing problem to limit the search for entering variables. The principal ideas are the following:

1. A list of the best entering variable candidates is formed, and pivots are made against that list until it is exhausted. (A pivot is the process of bringing in a new variable and rejecting an old one.)

2. When re-solving the problem during the routing process, it is often known where entering variables are most likely to be found: in a particular row or column, for example. This information is used to limit the search initially.

3. Channels are sorted so that the feasible columns in each row are grouped together; only that portion of each row needs to be searched.

4. As the routing proceeds and some demands are eliminated, those columns no longer need to be searched.

5.1.5.20 Subroutine SORTOF

This routine builds an ordered vector of basis elements on either rows or columns for use in finding a circuit in the basis during pivot operations.

5.1.5.21 Subroutine RCGEN2

The entire feasible region of the cost matrix is searched by this routine to find and store the best entering variable candidates.

5.1.5.22 Subroutine SORTRC

This routine sorts the list of candidates found by RCGEN2 by decreasing reduced cost.

5.1.5.23 Subroutine RCROW

A search is made in a limited range of rows for the best entering variable.

5.1.5.24 Subroutine RCCOL

A search is made in a limited range of columns for the best entering variable.

5.1.5.25 Subroutine DEPCHK

As described in Paragraphs 5.1.3.1 and 5.1.3.6, cargos that make intermediate stops end up in two separate channels: one going from the port of embarkation to the intermediate port; the other going from the intermediate to the port of debarkation. The latter channel is dependent upon the first, in that the first must complete its journey before the latter can begin. This routine is entered when a dependent channel is under consideration for placement on a ship. It's corresponding independent channel is examined to see if it has been placed on sailed ships in sufficient quantity to support the assignment under consideration.

5.1.5.26 Subroutine SECON1

SECON1 determines which channels are incompatible with the primary and marks them for elimination from future solutions (on the ship-trip currently being routed). Incompatibilities may stem from geography, improper ammunition loading, or improper Assault Follow-On Echelon loading. Geographical incompatibility is found by calculating the ratio of the trip distances, with and without the secondary channel, and comparing it with parameter PAR7.

5.1.5.27 Subroutine CHINIT

If the primary channel assignment is not a full shipload, secondary channel assignments will be considered for addition to the trip by simulating potential routes. CHINIT initializes the arrays and variables supporting these route simulations. The primary channel's ports are inserted into the trip's port list, interport distances are stored, a channel list is initialized, and two measures of performance are initialized: total distance traveled and expected lateness. If the expected lateness exceeds the maximum allowable lateness for this cargo type, the primary channel is rejected.

5.1.5.28 Subroutine CHEVAL

CHEVAL is used to evaluate a candidate secondary channel's compatibility with the ship trip under investigation. First, the channel's ports are added to the trip port list, and the resulting route is simulated. A measure of performance is calculated based on the expected lateness. Next, this measure is weighted by the number of alternative opportunities the candidate channel has available. (Fewer alternatives make the candidate more desirable.) The measure is also weighted by a ship fullness factor meant to improve ship utilization. Finally, the new measure of performance is compared with the measure before adding the candidate channel. If the difference is within the range specified by parameters PAR16, PAR17, and PAR18, the channel is accepted and delivery dates are retained for potential use in the cost function; otherwise, the trip port lists are restored to their previous states, and the candidate channel is marked as rejected.

5.1.5.29 Subroutine PORD

This routine adds a port to a list of ports to be visited by a ship when considering secondary channels. When possible, the insertion will minimize distance traveled. However, loading and unloading will be properly ordered and ammunition routing constraints will be followed. Ordered lists of ports, channels using those ports, interport distances, and estimated port departure times are maintained.

5.1.5.30 Function LODTIM

Given the ship, channel, stow type, and amount of cargo in the solution vector, LODTIM returns the time required to load the cargo.

5.1.5.31 Function UNLTIM

Given the ship, channel, stow type, and amount of cargo in the solution vector, UNLTIM returns the time required to unload the cargo.

5.1.5.32 Subroutine WOODEN

WOODEN determines if a ship compartment has been sheathed for ammunition loading and inserts the appropriate sheathing delay when needed.

5.1.5.33 Subroutine DONE

DONE scans the solution vector to discover whether all channels have either been assigned to the dummy ship or to sailed ships. If so, initial ship routing is complete. The routine is used only by the optimization-based option.

5.1.5.34 Subroutine QWAIT

Given the ship, stow type, channel assignment, and the time, QWAIT determines the expected queuing delay. The nominal queues were calculated in PODQ and POEQ in the

Aggregation Subsystem, so QWAIT needs only to determine the proper subscripts to retrieve the expected delay.

5.1.5.35 Subroutine COLSET

This routine places the costs in a given column of the cost matrix into a temporary vector.

5.1.5.36 Subroutine ROWGET

This routine finds the rows in the cost matrix corresponding to a given ship.

5.1.5.37 Subroutine ADDCHN

When a channel has been accepted for placement on a ship, this routine builds the temporary solution vector, first checking to see if the entire channel will fit. Actual stow factors are used. The amount of the ship's capacity used is incremented. ADDCHN is used only by the optimization-based routing option.

5.1.5.38 Subroutine BLDSOL

The temporary solution vectors built as the ships were being routed are collected into a single set of vectors by this routine.

5.1.5.39 Subroutine ACCEPT

This routine is called by subroutine CHEVAL when a channel is to be added to a ship's route. Its primary function is to mark the channel so that it will not be considered again on this ship trip; several other variables concerning the ship's trip are also updated.

5.1.5.40 Subroutine REJECT

When CHEVAL rejects a candidate channel, this routine is called to reset the route to the way it was before the channel was considered.

5.1.5.41 Subroutine OPTION

When a channel is being considered for addition to a ship, the following options are present: If it will be late in arriving, or if it will cause channels already on the ship to be late, the decision to add it or not depends on the number of alternative ships that might carry the candidate channel. This subroutine estimates the number of ship options available to the channel that are at least as good as the ship under consideration.

5.1.5.42 Subroutine TEST

This routine implements the decision process discussed in Paragraph 5.1.5.2 and illustrated in Fig. 5.1. Depending on how critical it is to add a channel, subroutine TEST will

decide whether or not too much additional lateness will be caused to channels already on the ship.

5.1.5.43 Subroutine OKNEO

This routine restricts ships from being routed to pick up noncombatant evacuees (NEO) as prime cargo if the ship is not within a given range of the port of embarkation. The range is controlled by parameter PAR15.

5.1.5.44 Function URAND

This routine generates random numbers between zero and one based on theory and suggestions given in D. E. Knuth (1969), vol. 2. A seed is accepted as an argument on the first call. The value of that argument should not be altered subsequent to the first call.

5.1.5.45 Real Function RISK2

This routine calculates the probability that a ship assigned to a route will suffer attrition. It accumulates the risk on each leg of the journey based on the port-to-port risks previously computed. (This is actually done by multiplying the probabilities that the ship does not sink; the complement of this product then gives the probability of attrition.)

5.1.5.46 Subroutine SHPSEL

This routine is used when heuristic routing is used to select a subset of ships for use in the optimization-based routing system. All ships used in the heuristic routing, plus certain high-value ships (whether used or not), are marked for inclusion in the setup of the optimization routing problem.

5.1.5.47 Subroutine MISMAT

This routine determines which ships cannot enter particular facilities at each port. For each facility, the capability of accommodating ships based on length, beam, and draft is compared to each ship. Unacceptable combinations of ship, port, and facility are marked.

5.1.5.48 Subroutine SOLU

This routine writes the current solution vector to the TESTOUT file for debugging. It is not routinely used.

5.1.6 Ship Loading Subsystem

The Ship Loading Subsystem begins with a detailed simulation of the Ship Routing Subsystem solution, which provides more precise dates for each ship's arrival and departure times at all ports on its route. These times can then be used to compute more accurate penalties for assigning particular channels to that ship.

With these better (more accurate) penalties, a new linear optimization problem is formulated in the general network format. This problem can represent individual cargo-to-ship stow factors exactly. It also gives the system a fresh start at the assignments, untainted by the iterative process used in the routing subsystem.

There are five segments to this subsystem: (1) computing arrival and departure times of ships at ports, (2) converting these times to costs, (3) constructing the general network problem, (4) solving the problem, and (5) interpreting the answers. Note that the fourth step, solving the problem, is accomplished using software written at the Georgia Institute of Technology, courtesy of Dr. Bill Nulty. Detailed descriptions of the solution subroutines will not be included in this report.

5.1.6.1 Subroutine ARRDEP

This program reads the LISTOUT file created in the simulation and determines the date of arrival of each ship in the ports visited. These dates are then used to determine costs (penalties) in the optimal loading problem.

5.1.6.2 Subroutine GNCDAT

This routine sets up the general network problem formulation for the optimal ship loading scheme. A schematic representation of the problem is given in Figure 5.2.

5.1.6.3 Subroutine GNC

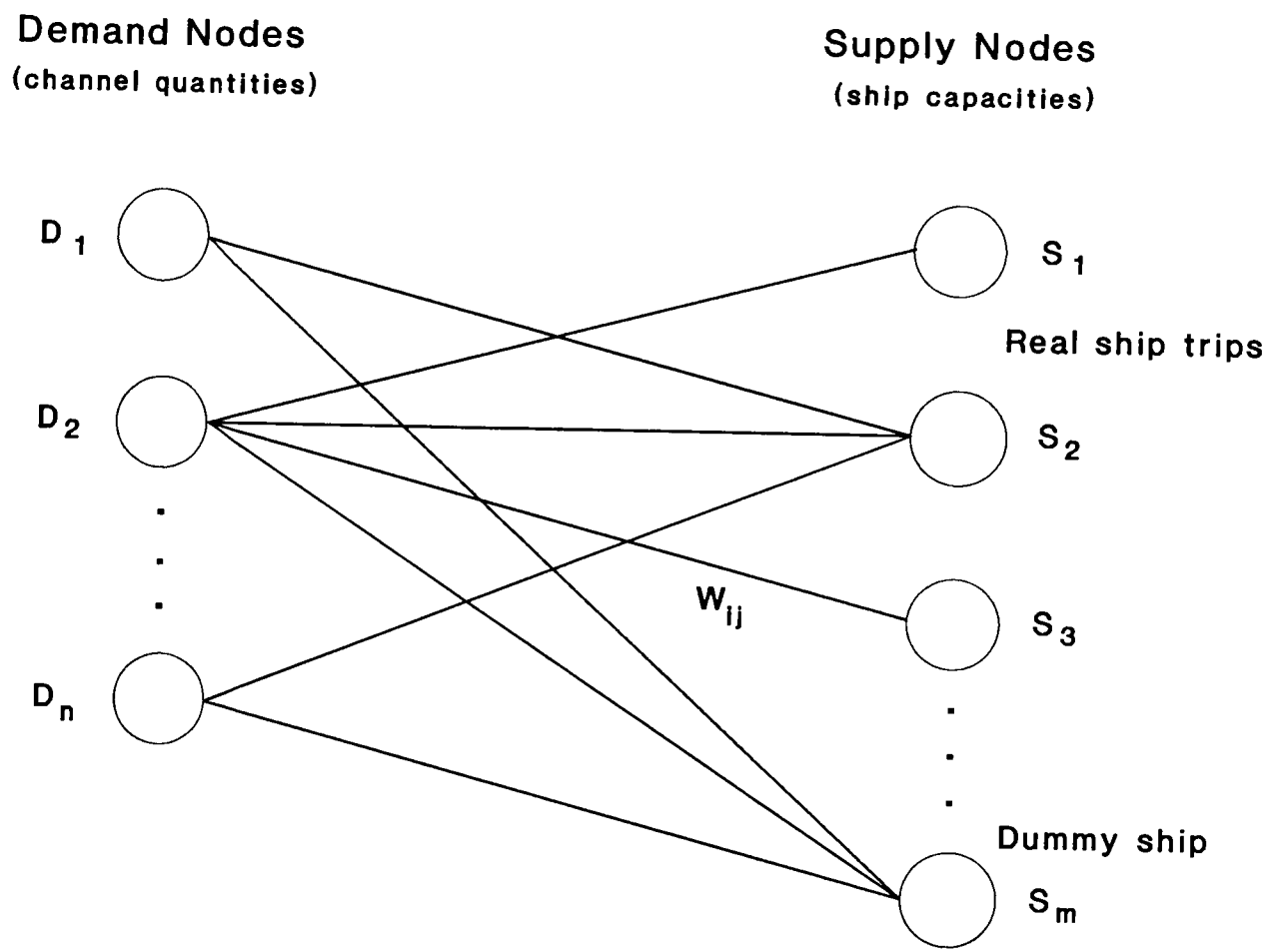
This is the interface routine between the scheduling code and the solution process for the general network algorithm used in the optimal ship loading process. The routine sets several parameters, calls subroutines ARRDEP and GNCDAT to set up the optimization problem, calls MGNC to solve the problem, and calls GNCOUT to put the solution back into a form which can be used by the rest of the SAIL routines.

5.1.6.4 Subroutine MGNC

This is the main control routine for the general network code. It calls the various subroutines required to reach an initial solution, test for optimality, and determine entering and leaving arcs.

5.1.6.5 Subroutines GFxxx

This set of 25 subroutines and 2 functions, controlled by MGNC, find a solution to the general network problem set up by subroutine GNCDAT. The individual routines will not be described here; their functions are well-commented in the code itself. These routines were provided by researchers at the Georgia Institute of Technology.



W_{ij}: Multipliers from Node D_i to Node S_j -- includes stow factor and, if necessary, square foot to measurement ton conversion

Fig. 5.2. The ship-loading problem - a generalized network formulation.

The inclusion of the stow factor as a multiplier on the flow in each arc assures that no compartment is overloaded, nor will it be unnecessarily underloaded. This multiplier also permits the capacity of dry cargo compartments to be represented by their correct units: square feet or measurement tons.

The costs are functions of how well the channel is served by the resource; as in the ship routing formulation, the cost is dominated by late delivery penalties. Arcs to the dummy ship naturally carry high costs. Parameters for the cost function are given in Paragraph 5.6.

5.1.6.6 Subroutine ENRICH

The ship routing solution prescribes a very specific route for each ship. In order to give the ship loading problem greater flexibility, it may be desirable to allow those routes to vary within controlled bounds. Introducing that variability is the role of subroutine ENRICH.

The enrichment of the problem is accomplished by using the concept of clustered channels, as defined in Paragraph 4.2.2. Channels are clustered together if they share certain attributes and have "nearby" ports of embarkation and debarkation ("nearby" is controlled by parameter PAR9; see Paragraph 5.2.3.2). Enrichment is achieved by the following process: the route from the ship routing subsystem permits certain channels to be carried; there are other channels clustered with those channels that the ship can carry, but which may use additional ports; those additional ports are added to the ship's route.

The actual mechanism for achieving this process is to add a small (one ton) load of these clustered channels to the solution vector of the ship loading process. Then the simulation (see Paragraph 5.1.6.1) will develop arrival and departure times for the added ports.

5.1.6.7 Real Function GTCOST

This function finds the cost, or penalty, for each arc of the general network problem. The basic approach is similar to that used in the setup of the transportation problem (see Paragraph 5.1.5.14); a hierarchy of criteria is established with two or more categories of quality for each criteria.

Cost functions are still subject to experimentation, so specifics may change. As of this writing though, the function contains three criteria in decreasing order of importance: a measure of the timeliness of delivery, a mechanism to limit the number of ships used, and at the lowest level, a measure of the compatibility of the cargo class and the compartment type.

The timeliness criteria is straightforward. Several categories of both lateness and earliness are defined; the costs in the most favorable category (on time delivery) are dominated by the costs in the next category (delivery a little early); early costs are in turn dominated by costs in the late categories.

The next level criterion, that of limiting the number of ships used, is a bit more difficult to explain. The ships are ordered into a ranking, best ships to poorest. (The rank is based on the ton-miles per day the ship can carry; actually, the way the ships are ranked

is of minor importance.) The use of a ship incurs a cost proportional to its rank (better ships have a lower rank number). This provision ensures that if two ships are serving a route, and both can deliver cargos with the same timeliness, one ship (the better) will always be filled before the other is used. (Obviously, both ships must be capable of carrying the cargo classes involved.)

The final criterion, cargo-compartment suitability, simply involves adding a penalty to the cost if the match is poor.

Figure 5.3 illustrates the process for a simple example involving three categories of timeliness and two ships. To read the figure, note that if the better-ranked ship carries a cargo with a poor stowage match, and delivers the cargo early, a cost of 6 units will be imposed. It can be seen in the figure how costs are constructed to insure that each level in the hierarchy of criteria dominates the level below it.

5.1.6.8 Subroutine SHRANK

This routine finds the rank of each ship based on its ton-miles-per-day carrying capacity. The best ships have the lowest rank numbers.

5.1.6.9 Subroutine GNCOUT

The results of the general network code are put into the standard solution vector form by this routine.

5.1.6.10 The General Network Solution Routines

The general network solution is accomplished by the routines shown in Table 5.2; the purpose of each routine is noted.

5.1.7 The Simulation Subsystem

This subsystem accepts the solution vector from either the Ship Loading Subsystem or the Ship Routing Subsystem and produces a detailed schedule of cargo loads and ship itineraries, including timing for all events. Two new files are produced: LISTOUT, an itinerary file that will be refined by the reporting subsystem; and DISTOUT, a file of ship routes and distances. Also, the file of cargos excluded from the schedule, EXCLOUT, is enlarged as unacceptably late cargos are removed.

There are two places in the SAIL model from which the Simulation Subsystem is called. The first is immediately following the Ship Routing Subsystem; the purpose is to obtain arrival and departure dates of ships at ports in order to compute better costs for the Ship Loading Subsystem. A variable, called PASS, keeps track of the number of passes that are made through the simulation system. PASS is set to zero for the first use of the simulation, and this signals the simulation programs to behave in a somewhat simplified manner: late cargos are not removed from the schedule, and control is returned to the calling program after a single pass.

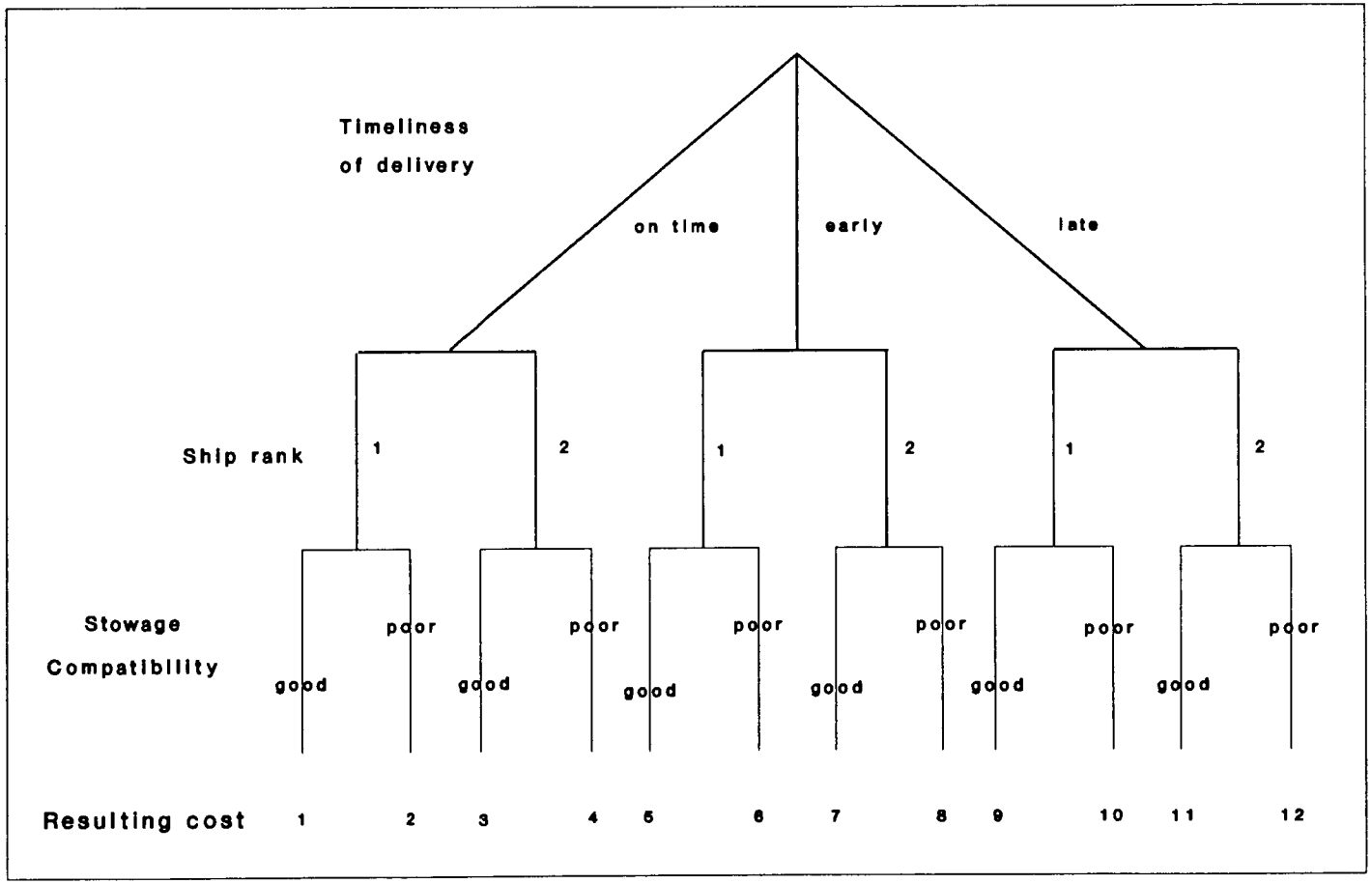


Fig. 5.3. Illustration of cost function for the ship-loading problem.

The second place that SAIL calls the subsystem is after the ship loading is completed. At this point, multiple passes through the simulation can be expected; sequential passes identify late cargos, attempt to reschedule those cargos on an earlier ship, and remove those cargos that remain unacceptably late. Generally, four or five passes are required to complete all of these functions. A final pass, arbitrarily numbered PASS=10, removes underutilized ships from the schedule, finalizes the timing of events, and writes the output files.

Table 5.2. General Network Subroutines

Subroutine	Function
GNC	Calls ARRDEP and GNCDAT to set up the GN problem and then call MGNC
MGNC	GNC control program
GFHSAV	Saves optimal flows in the basis
GFSART	Creates an artificial basis to get the algorithm started
GFPIV	Controls pivot routines
GFPDST	Calculates a dual variable
GFPDFD	Calculates dual value for a node when the dual for the node's predecessor is known
GFPDRV	Calculates dual value for a node when the dual for the node's successor is known
GFPSIN	Places an arc in the arc storage area
GFPSEX	Exchanges arcs from storage to basis
GFTUPS	Updates the subtree
GFTUPT	Updates predecessor and thread
GFTRRT	Re-roots a tree
GFTHNG	Hangs one tree on another
GFTPIN	Updates and isolates a subtree
GFTRCR	Takes a series of nodes along a cycle, reverses BPRED values, and puts them into a subtree
GFTCR	Takes a series of nodes along a cycle and puts them into a subtree
GFPIV2-7	These routines perform various types of pivots
GFPCHK	Given a node, determines if the associated arc is to leave the basis
GFPEXI	Given an entering arc, determines which arc will leave the basis
GFPFLW	Updates flows of basis arcs
GFPENT	Finds an entering arc
GFIARC	Returns information about an arc
GFOFLW	Updates flow on a single arc
GFHINV	Recalculates the flows and and value of the dual variables to decrease the chance of roundoff errors

Three variables associated with each ship--JNEXT(S), STAT(S), and TNEXT(S)--are key to the simulation. JNEXT(S) is the index number of the next event that ship S is to perform; it points to a load or unload of a given quantity of a particular channel (as determined from the solution vector). Intermediate stops are also potential events. STAT(S)

keeps track of the status of the ship. A variety of character codes indicate the ship's next activity: loading or unloading cargo, docking, sailing to a new port, or bunkering. A special code indicates the ship has completed all of its scheduled activities. TNEXT(S) contains the time that the next activity is to occur.

5.1.7.1 Subroutine SIMOVR

SIMOVR is a high-level control routine for the subsystem. It initializes variables, rewinds files as necessary, and controls the passes through the simulation. A logical variable, called RECYCL, is passed back to this routine from the simulations; if RECYCL is TRUE, another pass will be made through the simulation.

5.1.7.2 Subroutine SIMULA

SIMULA is a lower-level control routine. It begins with a series of calls to subroutines SKEDIN, INXTRA, ASSURE, and PATH, which are described in detail below. Basically, these routines read the solution vector and formulate the sequence of activities (loads and unloads of channels and bunker stops) which must be performed by each ship. The specific routes of the ships are found by PATH.

The simulation proceeds by considering the next activity to happen to any ship in the problem; that is, it finds the ship with the smallest value of TNEXT. That activity is explicitly simulated. For example, if the activity is a cargo loading, the ship's list of cargos is expanded; if it is an unloading, the list is reduced, and the cargo is registered as delivered; if it is a sailing, the time of arrival at the next port is found. In any event, the time the activity is completed is computed, the status indicators are updated appropriately, and the new value of TNEXT for that ship is found. A number of subroutines that accomplish these functions will be described below.

Simulation continues until all events are completed, or until some maximum time is exceeded. At the conclusion, summary information is written, and control is passed back to SIMOVR.

5.1.7.3 Subroutine SKEDIN

This routine translates the solution vectors into event-oriented arrays. These events are not ordered in time, however. The program proceeds as follows.

The solution vectors (from both the dry cargo and petroleum problems) are searched for real channels assigned to real ships. When an assignment is found, an event is established for the ship involved; the events are recorded by ship and trip. The information that must be retained for each event for a ship trip is the channel to be carried, the quantity of that channel, and the compartment of the ship into which the channel is to be placed.

The routine performs two other functions. For cargos that are unloaded at intermediate stops, it ensures that both legs of the trip are scheduled before setting up events for either leg. The program also builds some indexing vectors, the most important of which

is a list of cargo numbers sorted by channel number. These vectors are used in the loading process to load actual cargos (movement requirements) instead of the aggregated channels.

5.1.7.4 Subroutine INXTRA

This routine looks at the scheduled loading of each ship and computes the amount of unused space in each compartment. This space will subsequently be made available for rescheduling late cargo.

5.1.7.5 Subroutine ASSURE

As has been explained (see Paragraph 5.1.3.1), cargos that are unloaded at intermediate ports are treated by the optimization routines as two cargos. A considerable effort is made, using the penalty functions, to ensure that the two legs of the trip are properly sequenced. However, the proper relationship between these trips is not absolutely guaranteed. Subroutine ASSURE examines the solution, searching for improper sequencing and, if found, reassignments are made to regain feasibility.

5.1.7.6 Subroutine PATH

This routine finds the specific route for each ship to follow, and in so doing, establishes the order of the events identified in subroutine SKEDIN. Thus, the major input to the routine are the arrays developed by SKEDIN; the output is a new set of arrays which describe each ship's events in the proper sequence. To shorten the discussion that follows, we resort to the use of the acronyms POE, for port of embarkation, and POD, for the port of debarkation.

Of key importance in establishing a good route is finding the first port to visit. The rule used in PATH considers two ports as candidates for the first stop of any trip. One possibility is simply to start with the POE that is on the ship's list to visit on this trip and that is closest to the ship's current position. The other candidate is the POE that is farthest from the PODs on the trip. The rationale for this latter candidate is that the ship should collect its cargo on the way toward the PODs. The choice between these two candidates is made in the following way: if the candidate POEs are about equally distant from the PODs, the port closest to the ship is chosen; otherwise, the more distant port is visited first.

The remainder of the route is developed by the myopic rule of visiting the port closest to the current port, subject to several constraints. Obviously, POEs, intermediate ports, and PODs for a cargo must be visited in proper sequence. Also, there is an attempt made to visit each port on the route only one time, which may cause some deviation from the "next closest port" rule. And finally, if the ship is carrying both ammunition and nonammunition cargos, the ammunition must be loaded last and unloaded first.

5.1.7.7 Subroutine CENTER

This routine is called by PATH; it finds the centroid of the PODs to be visited on a trip and returns to path the POE farthest from that centroid.

5.1.7.8 Subroutine FAREST

This routine, called by CENTER, finds the POE farthest from the centroid of PODs.

5.1.7.9 Subroutine CHECK

This routine is called by PATH to see if a ship is carrying both legs of a channel with an intermediate stop. If so, the event index of the first leg of the trip is returned to PATH.

5.1.7.10 Subroutine REPATH

Subroutine PATH developed the sequence of port visits for each ship using a decision rule that can result in more miles being sailed than necessary. This routine examines the possibility of improving the route. All pairs of sequential visits are checked to see if reversing the order of the visit is possible (from the point of view of loading and unloading cargo in the correct order). If so, and if changing the order reduces the miles sailed, the exchange is made.

5.1.7.11 Subroutine STOPS

This routine finds the list of ports visited by a ship.

5.1.7.12 Subroutines SWPR and EXCH

When REPATH determines that the sequence of two ports visited by a ship should be reversed, subroutine SWPR switches the elements in the vector describing the port stops; subroutine EXCH exchanges the sequence vectors that contain the details of the ship's activity.

5.1.7.13 Subroutine PATHD

This routine finds the distance of a ship's route given a sequence of ports.

5.1.7.14 Subroutine CHKCH

The sequence of port visits cannot be switched if the same channel is involved at both ports; this routine is used by subroutine REPATH to check for this condition.

5.1.7.15 Subroutine CHKAMO

REPATH must also ensure that ammunition routing restrictions are not violated by a proposed reorder of port stops; this routine makes that check.

5.1.7.16 Subroutine EMBARK

When the next activity found for a ship in subroutine SIMULA is loading a channel, subroutine LOADUP is called. The pointer arrays built in SKEDIN are searched to determine which cargos are included in the channel. A cargo is selected from the list; the

selection is usually based on required arrival dates, earlier dates being favored, but intermediate stop constraints may also affect the selection.

The type of cargo and the type of ship compartment are examined to determine how much space the cargo will actually require. Cargos that exceed the available space are partially loaded, unless only a small amount of extra space is needed. The cargo is added to the ship's current list of cargos, and the amount of this particular cargo remaining to be loaded is decremented. Accounting is also done on port utilization, which is used by SIMULA to determine when a port is servicing ships at its capacity, thus causing queues of ships to develop. The time to load the ship is found, which permits updating the time of the ship's next activity (TNEXT).

If this is the last pass through the simulation system (i.e., if PASS=10), a record of the load is written on the LISTOUT file.

5.1.7.17 Subroutine DEBARK

When the activity in SIMULA calls for unloading a channel, subroutine DEBARK is called. The ship's cargo list is searched for cargos for which the present port is a POD. Those cargos are unloaded, and the time is compared with the latest acceptable delivery time. If a cargo is unacceptably late, two things happen: first, the variable RECYCL is set to TRUE, indicating the simulation will need to be repeated; and second, variables related to the cargo, including the solution vector, are changed so the cargo will not be loaded on the next pass through the simulation. The time required for the unloading is found, permitting the ship's next event time to be updated. Finally, the cargo is removed from the list of cargos on the ship.

Again, when PASS is equal to 10, a record of the unloading is written on the LISTOUT file.

5.1.7.18 Subroutine STATUS

This routine is called by SIMULA to determine the ship's status; that is, whether the next activity of the ship is a load, unload, bunkering, docking, or sailing. It also determines when the ship is finished with its activities.

5.1.7.19 Subroutine UNDEL

When the simulation is completed, this routine places any remaining (undelivered) cargos on the EXCLOUT file.

5.1.7.20 Subroutine PORTTM

The time a ship requires for a loading or unloading function at a port is computed by this routine. For the period of time required, it is assumed that the operation will absorb port throughput capacity at the ship's load or unload rate for the class of cargo involved. Arrays that keep track of the amount of port throughput remaining are updated; these arrays are

used to determine if other ships can dock, or if they must wait for the facility to become available.

5.1.7.21 Subroutine CHKCAP

This routine checks the availability of port throughput capacity prior to docking a ship. If none is available, an expected available time is computed and the ship is delayed until that time.

5.1.7.22 Subroutine INSERT

This routine inserts an event time in the port use arrays. This is used by PORTTM in the process of accounting for port utilization which, in turn, determines whether or not ships may dock or must be queued.

5.1.7.23 Subroutine ADEQAT

This routine checks to see which ship-trips are adequately loaded by the minimum ship load criterion. Trips that have enough cargo on board are marked for sailing; others are marked for elimination.

5.1.7.24 Subroutine SAILIN

This routine initializes the sailing indicator array set by subroutine ADEQAT.

5.1.7.25 Subroutine FILLCP

When port throughput capacities are changing over time, this routine fills in the appropriate arrays from the date of change onward. It also checks to see that the maximum number of changes has not been exceeded and that the dates of change are in increasing order.

5.1.7.26 Subroutine PORCHK

Each cargo is checked to see that an appropriate port facility exists; if no facilities are found, the cargo is eliminated and a message is written on the EXEPOUT file.

5.1.7.27 Subroutine SRTULL

The activities at a port are sorted so that all unloads at a facility occur before any loads at that same facility.

5.1.7.28 Subroutine CHANGE

This routine switches two elements being sorted by subroutine SRTULL in the vectors.

5.1.7.29 Subroutine NOSTOP

The objective of this routine is to eliminate port stops that do not involve adequate amounts of cargo, as defined by parameter PAR11 for ports of embarkation and parameter PAR12 for ports of debarkation. The sequence of events planned for each ship is scanned, and the total activity of each port is computed. The operations at ports that do not meet the minimum criterion are eliminated. Since this may affect the activity at other ports, the process must be repeated until no change occurs.

5.1.7.30 Subroutine ENUF

Subroutine NOSTOP uses this routine to determine if the level of activity at a port is sufficient.

5.1.7.31 Subroutine REMOVE

Subroutine NOSTOP uses this routine to remove operations from a ship with insufficient activity at a port.

5.1.7.32 Subroutine NXCSQF

When loading individual cargos on a ship, this routine finds the next cargo to load out of a channel for stowing in compartments measured in square feet.

5.1.7.33 Subroutine NXCP

This routine finds the next cargo to load out of passenger or POL channels.

5.1.7.34 Subroutine NXCMT

This routine finds the next cargo to load out of measurement ton channels.

5.1.7.35 Subroutine OSCOMP

To save memory, the arrays which contain information about cargos on ships are stored using a pointer system; when the pointer runs out of room, the arrays are compressed to get rid of cargo that has been unloaded. This routine accomplishes that compression.

5.1.7.36 Subroutine FINDCP

Since port throughput can change over time, this routine is used to find the throughput of a particular port facility on a given c-day.

5.1.7.37 Subroutine SLPCHK

An effort is made in the simulation process to avoid calling a ship into service unnecessarily early. The ship is "slipped" to a later date to the extent possible and still avoid late deliveries. Clearly, if a ship is late on its last trip, its schedule on its first trip should not

be "slipped" later, regardless of the earliness of the first trip. Subroutine SLPCHK assures that the amount of slippage is not higher for an earlier trip than it is for a later one.

5.1.7.38 Subroutine STATUS

During the simulation process, this routine returns the next status of a ship; that is, whether the next operation is a load, unload, sailing, or the completion of all activity.

5.1.7.39 Subroutine ALDCHK

This routine examines all the cargo loads to be performed at a port and delays the ship's docking until the loading can proceed without waiting for cargos to arrive in the port.

5.1.7.40 Subroutine ENDSIM

This routine wraps up the simulation process and writes out some final information about the simulation process. It also builds the ship risk table based on the attrition data provided in the RISKDAT file.

5.1.8 The Refinement Process

Associated with the Simulation Subsystem is an optional process to refine the loads on the ships. This process, invoked by the parameter DOREEF, seeks to reassign cargos that arrive late (or that are unscheduled) to different ships (see Paragraph 4.3.22). The following routines implement this strategy.

5.1.8.1 Subroutines REFIN1 and REFIN2

The purpose of these routines is to refine the assignments of channels to ships after simulation has revealed the actual delivery dates and ship utilizations. The basic approach is to select a ship that could accept more cargo and then determine if a significant amount of undelivered or late cargo could be transferred to it. If such a transfer is possible, and if lateness is diminished, the transfer is made.

The difference in the two routines is that, in REFIN1 the ships are considered beginning with the most used ships, while REFIN2 operates in the reverse order. The objective in the first case is to load up the better-used ships prior to eliminating ships for sparse loads. After the underutilized ships are eliminated, the reverse search looks for new opportunities for improving the timeliness of deliveries.

5.1.8.2 Subroutine DUMSHP

The ship loading solution formed by subroutine GNCOUT does not contain elements representing the uncarried cargo; that is, channels scheduled on the dummy ship. Subroutine DUMSHP restores these elements to the solution vector for use by the refinement routines.

5.1.8.3 Integer Function LV

This function finds the time that a ship leaves a particular port on a given trip. These times have been stored in an array by the simulation process.

5.1.8.4 Subroutine INIMRK

This routine initializes marker vectors used by the refinement routines.

5.1.8.5 Subroutine COMPAT

When considering the reassignment of a channel to a different ship, this routine checks the compatibility of the cargo class, ship compartment type, and port facilities. A flag is set if the proposed reassignment is infeasible.

5.1.8.6 Subroutine NEWROW

This routine finds the row values corresponding to a ship (and trip) chosen to receive reassigned cargo.

5.1.8.7 Subroutine FRACLD

The fraction of a full load is found for each ship.

5.1.9 The Bunker Routines

Also associated with the Simulation Subsystem are the bunkering routines that implement the process described in Paragraph 4.3.20.

5.1.9.1 Subroutine FUELS

The ships are tracked on their routes, and the fuel consumption is monitored. Bunkering is performed whenever required to maintain the specified minimum fuel load. If the route does not include suitable bunker ports, a diversion to a bunker port is added, if possible. Ships that cannot continue because of fuel restrictions are marked to prevent them from continuing their trip.

5.1.9.2 Subroutine PITSTP

When subroutine FUELS determines that a diversion to a bunker port is necessary, this routine is called to find the port to divert to and insert this port into the ship's route.

5.1.9.3 Subroutine FILL

As fuel is loaded onto ships, this routine increments the amount of fuel used (by type) at the port and sets the ship's fuel indicator to full.

5.1.10 The Reporting Subsystem

The primary purposes of the Reporting Subsystem are the following: (1) to analyze the slack in the schedule and remove earliness where possible; (2) to merge the excluded cargo with the scheduled cargo; (3) to display summary statistics of the results; and (4) to write the results on file FINIOUT for subsequent processing by the SEASTRAT reporting system.

Although SAIL has no user reporting responsibilities, cargo delivery histograms and several reports are produced by subroutine POST for testing and comparison purposes.

5.1.10.1 Subroutine POST

To accomplish the subsystem's goals, POST first reads from file LISTOUT the scheduled activities created in SIMULA and from file EXCLOUT the unscheduled cargos accumulated during the input, the routing, and the simulation phases. The schedule is scanned in reverse order to determine the real slack time available at each activity. Two additional schedules are created using this new information. The first of these moves cargo delivery to the latest possible time without creating any new delivery lateness. The second moves deliveries only as far as the early arrival date. The latter of these sets of dates is used as the primary date on file FINIOUT.

Summary reports of ship utilization, delivery by cargo class, and port activity similar to reports from SEACOP (an existing planning model) are produced. Also, histograms showing the amount of cargo delivered by days early, days late, and on-time are created for each cargo class.

5.1.10.2 Subroutine REPTC

This routine writes a report of percentages of cargo on-time, late, and unscheduled by cargo class on the HISTOUT file.

5.1.10.3 Subroutine REPTS

This routine writes on HISTOUT the number of ships available for this run, the number of ships used, the number of trips, the total mileage (in nautical miles), the average mileage per trip, and the average compartment stow-adjusted loading for each ship type.

5.1.10.4 Subroutine REPTP

REPTP produces a summary port activity report on HISTOUT including: total requirements, amount scheduled and unscheduled as a port of embarkation, amount scheduled and unscheduled as a port of debarkation, and total shortfall by port type.

5.1.10.5 Subroutine REPTSC

This routine produces a report on the HISTOUT file which shows the percentage of cargo of various classes carried by each ship type. It also gives the percentage of each ship

type which was used to carry each of the cargo classes. Only cargo classes requested by planning analysts are represented in the report.

5.1.10.6 Subroutine STATS

Statistics on delivery lateness and earliness and infeasible cargos are collected by this routine.

5.1.10.7 Subroutine CARSTO

A table is written on the HISTOUT file showing the amount and class of cargo carried in the various compartments of all ship types used in the plan.

5.1.10.8 Subroutine HISTO

This routine prints the histograms of lateness and earliness on the HISTOUT file.

5.1.10.9 Subroutine PWRREP

This routine writes the refueling report on the HISTOUT file.

5.1.10.10 Subroutine LCREP

For lift capability assessments, a capability report is written on the HISTOUT file. Five nominal cargo types are reported: breakbulk, container, vehicles, passengers, and petroleum; the amount of cargo of each class that can be delivered over each channel is shown in 10-day increments.

5.1.10.11 Subroutine RLREPT

For a required lift analysis, this routine writes a report on the HISTOUT file. For each port of embarkation, the report shows the number of nominal ships of each type required by 10-day time intervals.

5.1.10.12 Subroutine RISKCS

Risks for both ships and cargos are accumulated by this routine. This permits reporting the probability that a ship or cargo will be lost; cargo loss probabilities are reported on the FINIOUT file.

5.1.11 Subroutines to Support the Transfer Port Concept

Operations using a transfer port were introduced in Paragraph 4.3.28. The additional subroutines necessary to implement the concept are described here. The transfer port concept uses the SAIL algorithm with only a few modifications; actually, two passes are made through most of the scheduling process. On the first pass, the cost function is used to assure the restricted ships are not allowed to carry cargo to the forbidden ports; also, cargos which have adequate slack time in their schedules are given the transfer port as an intermediate

stop. Additionally, the variables that control the maximum allowable lateness are set to small numbers; this reduces the opportunity for scheduling cargos to the transfer port which cannot be realistically delivered on to their final destination on-time. The result is that the restricted ships will be used, to the extent possible, to carry cargo to the transfer port.

But the outcome, at this point, is unsatisfactory because all of the cargos with adequate slack time were forced through the transfer port, whether they were carried by the restricted ships or not. This leads to a second pass through the algorithm. The cargos that were carried by restricted ships are assigned to those ships using the preferred ship concept. All other cargos are returned to their original status...that is, they no longer are routed through the transfer port. With these initializations, a normal pass is made through the SAIL process.

The result is a schedule that has some cargo being carried to the transfer port on the special ships and then continuing on unrestricted ships to their final port of debarkation. All other cargos are carried on regular ships.

5.1.11.1 Subroutine SETPOX

This is a control routine for reading information about the transfer ports and restricted ships and setting up the data for the first pass through the algorithm.

5.1.11.2 Subroutine INPOX

Two files are read. The POXPDAT file contains the transfer port GEOLOC code followed by the GEOLOC codes of all the ports forbidden to the special ships. The POXSDAT file contains a list of the special ship's NISC numbers.

5.1.11.3 Subroutine POXSAV

This routine saves certain information about the cargos prior to making the first pass so the cargo characteristics can be restored prior to the second pass. For example, the cargos which have adequate slack in their schedule will be re-routed using the transfer port as an intermediate stop. This routine stores the original information for restoration if the cargos are not actually routed that way on the first pass.

5.1.11.4 Subroutine FORBDN

There is an array called MISPAT that indicates allowable combinations of ships and ports. Using the MISPAT array, this routine assures that the special ships will not be allowed to enter the ports forbidden to them.

5.1.11.5 Subroutine POXLAD

For those cargos with adequate slack, such that they are candidates for routing through the transfer port, this routine finds latest arrival dates at the transfer port. The objective is to find a date so that the cargo has a reasonable chance of being scheduled on to its actual port of debarkation before the original arrival date.

5.1.11.6 Subroutine RESTOR

After the first pass through SAIL, this routine restores the cargo data and maximum lateness variables to their original values.

5.1.11.7 Subroutine MAKEPOI

For cargos actually carried by special ships in the first pass, the cargo's intermediate port field is set to the transfer port and the preferred ship field is set to the special ship that actually carried it. If a cargo was only partly routed through the transfer port, the cargo record is split to represent both routing processes. (The routine also checks for cargos for which the transfer port is the final destination.)

5.1.11.8 Subroutine SETSAD

This routine resets the ship availability dates between the first and second passes through the SAIL scheduler.

5.1.12 Lift Capability Analysis

Special routines are needed to assess the capability of available lift. These routines set up special requirements based on the defined channels and perform a simplified routing algorithm for this problem.

5.1.12.1 Subroutine LCINPT

Channels for lift capability assessments are read by this program. A channel indicates a port of embarkation, port of debarkation, type of cargo moving, and the fraction of the total movement represented by the channel.

5.1.12.2 Subroutine LCYCLE

This is a special control program for routing ships for lift capability assessments; it is analogous to programs NCYCLE and ACYCLE used in heuristic and optimal-based routing.

5.1.12.3 Subroutine LPRIME

For lift capability assessments, this program finds the next most critical channel. The selection is based on the difference between the fraction of cargo flowed over a channel from the targeted fraction.

5.1.12.4 Subroutine LSAILM

This is the lift capability equivalent to subroutines NSAILM and ASAILM; it fills in the solution vector for the sailed ship and determines when the ship will be available for another sailing. If the future sailing date is beyond the time period specified for the lift capability assessment, the cost function is modified to prevent the ship's reuse.

5.1.12.5 Subroutine LSTASH

Similar to subroutine STASH, this program stores a channel on a ship. In lift capability computations, however, there is no fixed amount to be stored; the routine places as much materiel on the ship as will fit.

5.2 INPUT AND OUTPUT FILE STRUCTURES

In this section, the interface between the SAIL model and the remainder of the SEASTRAT system is defined. The mechanism for this interface is a set of input and output files. The section begins with a map describing the subroutines that open, use, and close each file; additional paragraphs tabulate the structure and content of the files. Throughout this section, the field names in each file closely conform to the naming conventions used by other SEASTRAT developers.

5.2.1 Blocks of Code and the Subsystems

The subroutines of SAIL are assembled into blocks of code for editing and management purposes; the blocks correspond to the subsystems as follows:

SUBSYSTEM	BLOCK
Interface	MAIN, INPUT1, INPUT2
Aggregation	CHANEL
Distance	DIST1, DIST2, ATTRIT
Ship Routing	CYCLE, MATGEN, SOLVE, OPTA, and NOPT
Ship Loading	LP84 and GENNET
Simulation	S1, S2
Embarking/Debarking	BARK
Routes	PATH
Refine	REFINE
Bunkering	BUNKER
Reporting	POST
Utility Programs	UTIL
Write Data Edits	
Write Solution Vector	
Distances	
Infeasible Ship/Port/Cargos	
Simulation	

5.2.2 File Usage Table

Table 5.3 lists all files used by SAIL and shows the blocks of code that access them. The FORTRAN file number associated with each file is also given.

Table 5.3. SAIL File Map

FORTRAN File No.	File Name	Subsystem Block	Open	Read	Write	Close
1	HISTOUT	MAIN	Y			
		S2			Y	
		POST			Y	
2	RNAME(PHASE)	ATTRIT	Y	Y		Y
2	CHANOUT	CHANEL	Y		Y	Y
2	CRGODAT	INPUT1	Y	Y		Y
2	SHPTDAT	INPUT2	Y	Y		Y
2	SHIPDAT	INPUT2	Y	Y		Y
2	PORTDAT	INPUT2	Y	Y		Y
2	COMPDAT	INPUT2	Y	Y		Y
2	PREFDAT	INPUT2	Y	Y		Y
2	THRUDAT	INPUT2	Y	Y		Y
2	LOADDAT	INPUT2	Y	Y		Y
2	UNLDDAT	INPUT2	Y	Y		Y
2	STOWDAT	INPUT2	Y	Y		Y
2	FACIDAT	INPUT2	Y	Y		Y
2	PARADAT	INPUT2	Y	Y		Y
2	PLANDAT	INPUT2	Y	Y		Y
2	LCAPDAT	INPUT2	Y	Y		Y
		POST	Y	Y		Y
2	NODEDAT	DIST1	Y	Y	Y	
2	LINKDAT	DIST1	Y		Y	Y
2	SOLSAV=	MAIN	Y	Y		Y
2	SOLGET=	MAIN	Y		Y	Y
2	DISSAV=	DIST1	Y		Y	Y
2	DISGET=	DIST1	Y	Y		Y
2	SADSAV=	DIST1	Y		Y	Y
2	SADGET=	DIST1	Y	Y		Y
2	LINEDAT	DIST1	Y	Y		Y
2	INTPORT	DIST1	Y		Y	Y
2	SADSOUT	DIST1	Y		Y	Y
2	VIEWDAT	DIST2	Y	Y		Y
2	BYPASS	REUSE	Y		Y	Y
13	EXEPOUT	MAIN	Y		Y	
		ALL			Y	
16	TESTOUT	MAIN	Y		Y	
		ALL			Y	
17	EXCLOUT	MAIN	Y		a	
		CHANEL			a	
		CYCLE			a	
		SIMULA			a	
18	LISTOUT	MAIN	Y			
		BARK			Y	
		POST		Y		
19	FINIOUT	MAIN	Y			
		POST			Y	
20	DISTOUT	MAIN	Y			
		S1			Y	
		S2			Y	

[a]All writes to this file are done through Subroutine WREXCL, which is in the UTIL block.

5.2.3 Description of Input Files

The input files may be divided into three types: the major data files for cargo, shipping resources, and port descriptions; the parameter files; and semipermanent data that describe ship types, loading rates, stow factors, and the ocean network used in distance computations. In these descriptions, the field description, format, and internal SAIL variable name are given. The input data are often altered before storing; in these cases, the internal variable is shown in parentheses.

5.2.3.1 Major Data Files

The following three files contain the heart of a plan's input; they define the requirements to be moved, the ships available to provide the lift, and the ports involved in the plan. These files, along with the parameter files (PARADAT and PLANDAT) are the parts of the input that always change from analysis to analysis. It is essential that these files be complete and properly edited prior to presentation to the SAIL algorithm for the results to be comprehensive. SAIL will make some edits on the data, but, for the most part, invalid records must be discarded; the resulting analysis is therefore incomplete. (In some cases, invalid data will cause SAIL to halt entirely.)

CRGODAT

Each record of this file describes an individual movement requirement. The records begin with 12 characters of plan identification, followed by a unique record identification number. It is this latter number that is used to tie results, as given in the FINIOUT file, back to the original data:

Field Description	Internal Format	Column Variable	Number
OPLAN	A6		1-6
CYCLE	A2		7-8
RUN	I3		9-11
MODULE	A1		12
REQUIREMENT ID	I6	RID	13-18
CARGO CLASS (see Paragraph 5.5)	A2	(CTYPE)	19-20
ORIGIN GEOLOC CODE	A4	OGEO	21-24
POE GEOLOC CODE	A4	(CPOE)	25-28
INTERMEDIATE GEOLOC CODE	A4	(CPOI)	29-32
POD GEOLOC CODE	A4	(CPOD)	33-36
ORIGIN READY TO LOAD DATE	I3	RLD	37-39
POE AVAILABLE TO LOAD DATE	I3	ALD	40-42
INTERMEDIATE DAYS DELAY	I3	DDELAY	43-45
POD EARLIEST ARRIVAL DATE	I3	EAD	46-48
POD LATEST ARRIVAL DATE	I3	LAD	49-51
PASSENGERS	I6	(CQTY1)	52-57
SHORT TONS	I6	STONS	58-63
MEASUREMENT TONS	I6	(CQTY1)	64-69
SQUARE FEET	I6	(CQTY2)	70-75
PETROLEUM, OIL, AND LUBRICANTS (POL) in CBBLS	I6	(CQTY1)	76-81
CARGO PORT TYPE (A,B,C,G,P)	A1	(CFAC)	82
FUEL TYPE	A3	FUELTP	83-85
PERFERRED SHIP	I5	(PSHIP)	86-90
PERFERRED SHIP TYPE	I2	(PSHIPT)	91-92
SCHEDULING PRIORITY	A1		93
REASON FOR INTERMEDIATE STOP	A1		94
SHIP CONSTRAINTS	A1		95
DISCHARGE CONSTRAINTS	A6		96-101
REQUIREMENT CONSTRAINTS	A1	REQCON	102
POE COMPLEX	A5		103-107
UNIT	A6	UNIT	108-113
LOAD DATE	I6		114-119
LOAD TIME	I4		120-123

TOTAL LENGTH OF ALL FIELDS = 123 CHARACTERS

SHIPDAT

The SHIPDAT file contains the description of the ships that are available for use in the plan, along with their location and current destination:

Field Description	Internal Format	Column Variable	Number
OPLAN (Optional)	A6		1-6
CYCLE (Optional)	A2		7-8
RUN (Optional)	I3		9-11
MODULE (Optional)	A1		12
NISC NUMBER	I5	SID	13-17
FLEET CODE (see Section 5.6)	I2	FLEET	18-19
SHIP TYPE (see Section 5.5)	I2	(STYPE)	20-21
SHIP SPEED (knots)	F4.1	(SPEED)	22-25
SHIP LENGTH (feet)	I4	SLENG	26-29
SHIP BEAM (feet)	I3	SBEAM	30-32
SHIP DRAFT (feet)	I2	SDRAFT	33-34
BOOM CAPACITY	I4	BOOM	35-38
SHIP FUEL TYPE	A3	FUEL	39-41
SHIP FUEL CAPACITY (long tons)	I6	FUELCP	42-47
SHIP FUEL CONSUMPTION (tons/day)	I4	FUELUS	48-51
SHIP B/B CAPACITY (measurement tons)	I6	CAP	52-57
SHIP CONT CAPACITY (measurement tons)	I6	CAP	58-63
SHIP RO/RO CAPACITY (square feet)	I6	(CAP)	64-69
SHIP LO/LO CAPACITY (square feet)	I6	(CAP)	70-75
SHIP POL CAPACITY (cbbls)	I6	CAP	76-81
SHIP PAX CAPACITY (persons)	I6	CAP	82-87
ADMINISTRATIVE DELAY DAYS	I3	ADDS	88-90
EARLY WARNING DAYS	I3	EWDS	91-93
SELF SUSTAINING FEATURE	A1		94
TANK CLEANING DELAY (days)	I3	SPREP	95-97
ACTIVATION PREPARATION	I3	ACTPRP	98-100
PRESENT LATITUDE DEGREES	I2	(SPLAT)	101-102
PRESENT LATITUDE MINUTES	I2		103-104
PRESENT LATITUDE HEMISPHERE	A1	(SPLAT)	105
PRESENT LONGITUDE DEGREES	I3	(SPLON)	106-108
PRESENT LONGITUDE MINUTES	I2		109-110
PRESENT LONGITUDE HEMISPHERE	A1	(SPLON)	111
DESTINATION LATITUDE DEGREES	I2	(SDLAT)	112-113
DESTINATION LATITUDE MINUTES	I2		114-115
DESTINATION LATITUDE HEMISPHERE	A1	(SDLAT)	116
DESTINATION LONGITUDE DEGREES	I3	(SDLON)	117-119
DESTINATION LONGITUDE MINUTES	I2		120-121
DESTINATION LONGITUDE HEMISPHERE	A1	(SDLON)	122
FIRST DEDICATED PLAN (if any)	A12	(SOPLAN)	123-134
SECOND DEDICATED PLAN (if any)	A12	(SOPLAN)	135-146
THIRD DEDICATED PLAN (if any)	A12	(SOPLAN)	147-158
IN-PROCESS TRIP NUMBER*	I2	PTRIP	159-160
THE DAY IN-PROCESS TRIP BEGAN*	I3	PWHEN	161-163
THE NUMBER OF TRIPS THE SHIP CAN MAKE	I2	NRTRIP	164-165
LOAD DATE	I6		166-171
LOAD TIME	I4		172-175

TOTAL LENGTH OF ALL FIELDS = 186 CHARACTERS

*To be used in execution planning.

PORTDAT

The PORTDAT file describes all ports used in the plan. The throughput capacity and dimensions of each of five facility types are given; the five types are

A: General Ammunition
B: Container Ammunition
C: Container
G: General
P: Petroleum

The location of the port, the availability of bunker facilities, and information regarding queuing and AFOE considerations are also given. The port is identified by its geolocation code.

It should be noted that there is an additional data element needed in the PORTDAT file. To fully implement the limitation on the use of Korean and NATO ships (see Paragraph 4.3.1), SAIL must know the country associated with each port. This field needs to be added to the PORTDAT records. SAIL already has an array, ASSOC(port), that associates ports with various regions; this array should receive its information from the new PORTDAT field.

Field Description	Internal Format	Column Variable	Number
OPLAN	A6		1-6
CYCLE	A2		7-8
RUN	I3		9-11
MODULE	A1		12
FACILITY GEOLOC	A4	PID	13-16
PORT COMPLEX	A5	COMPLX	17-21
PORT LATITUDE DEGREES	I2	(PLAT)	22-23
PORT LATITUDE MINUTES	I2	(PLAT)	24-25
PORT LATITUDE HEMISPHERE	A1	(PLAT)	26
PORT LONGITUDE DEGREES	I2	(PLONG)	27-29
PORT LONGITUDE MINUTES	I2	(PLONG)	30-31
PORT LONGITUDE HEMISPHERE	A1	(PLONG)	32
AFOE RETURN DATE	I3	AFOERD	33-35
AFOE OFFSET DAYS	I2	AFOEID	36-37
BUNKERING INDICATOR ('Y' activates)	A1	(PBUNK)	38
PORT AVAILABILITY DATE	I3	PAD	39-41
QUEUEING INDICATOR ('Y' deactivates)	A1	(PQUE)	42
PORT TYPE A			
PORT TYPE	A1	PTYPE	43
PORT LENGTH (feet)	I4	PLTH	44-47
PORT BEAM (feet)	I3	PBEAM	48-50
PORT DRAFT (feet)	I3		51-53
PORT OVERDRAFT (feet)	I3	PDRAFT	54-56
MILITARY WARTIME THROUGHPUT (MT/day)	I7	MCAP	57-63

PORTDAT (continued)

Field Description	Internal Format	Column Variable	Number
PORT TYPE B			
PORT TYPE	A1	PTYPE	64
PORT LENGTH (feet)	I4	PLTH	65-68
1PORT BEAM (feet)	I3	PBEAM	69-71
PORT DRAFT (feet)	I3		72-74
PORT OVERDRAFT (feet)	I3	PDRAFT	75-77
MILITARY WARTIME THROUGHPUT (MT/day)	I7	MCAP	78-84
PORT TYPE C			
PORT TYPE	A1	PTYPE	85
PORT LENGTH (feet)	I4	PLTH	86-89
PORT BEAM (feet)	I3	PBEAM	90-92
PORT DRAFT (feet)	I3		93-95
PORT OVERDRAFT (feet)	I3	PDRAFT	96-98
MILITARY WARTIME THROUGHPUT (MT/day)	I7	MCAP	99-105
PORT TYPE G			
PORT TYPE	A1	PTYPE	106
PORT LENGTH (feet)	I4	PLTH	107-110
PORT BEAM (feet)	I3	PBEAM	111-113
PORT DRAFT (feet)	I3		114-116
PORT OVERDRAFT (feet)	I3	PDRAFT	117-119
MILITARY WARTIME THROUGHPUT (MT/day)	I7	MCAP	120-126
PORT TYPE P			
PORT TYPE	A1	PTYPE	127
PORT LENGTH (feet)	I4	PLTH	128-131
PORT BEAM (feet)	I3	PBEAM	132-134
PORT DRAFT (feet)	I3		135-137
PORT OVERDRAFT (feet)	I3	PDRAFT	138-140
MILITARY WARTIME THROUGHPUT (cbbls/day)	I7	MCAP	141-147
LOAD DATE	I6		148-153
LOAD TIME	I4		154-157

TOTAL LENGTH OF ALL FIELDS = 157 CHARACTERS

5.2.3.2 Parameter Files

Many SAIL heuristics can be controlled via parameters that can be set external to SAIL and accepted by SAIL as input files. There are three parameter files. PLANDAT contains items under the planning analyst's control. PARADAT contains items currently reserved for use by system developers or that are set by MSC policy. VIEWDAT contains information about the "view" of the world: the status of canals, when conditions are expected to change, and the names of files where risk and ocean avoidance data are stored. These files are prepared by SEASTRAT's Database Management System and passed to the SAIL algorithm.

PLANDAT

Although the PLANDAT file is described in detail in *SEASTRAT Planner Parameter File Reference Document*, for completeness a short discussion of this file will be included.

Some of the prior functions of the PLANDAT file have been taken over by the VIEWDAT file; also, PLANDAT is now the source of information which permits multiple OPLAN analyses to be made.

The first record of the file limits the lateness allowed for the various cargo categories; there is currently no provision to make these limits plan-dependent. Maximum lateness is given in days; if a cargo cannot be delivered within the specified number of days of the latest delivery date, the cargo will be removed from the schedule. There are five categories of cargo: unit equipment or personnel, replacement personnel, resupply ammunition, general resupply, and resupply petroleum. The fields for these entries are integers, three columns each.

Following the maximum lateness on the first record are the following grab-bag of run parameters:

A one-character parameter indicates whether or not AFOE cargos can be mixed with other cargos; a "Y" indicates mixing is permitted. **This feature is not currently used; AFOE is never mixed.**

A one-character parameter indicates whether or not ammunition can be mixed with non-ammunition cargos; a "Y" permits mixing.

The next field determines whether the conditions are peacetime or wartime. This affects the initial availability of ships. A "W" indicates war; a "P" indicates peace.

The nominal time to refuel a ship in days is next; this is a real number of width three (F3.1).

The last ten digits of the record contain the time-date stamp supplied by SEASTRAT; they are not used by SAIL.

The second and subsequent records of PLANDAT tell which plans are being scheduled and when each begins to execute. The first 12 characters contain a plan identification; the next four positions give the offset (in days) from the beginning date of the scenario. That is, if a particular plan is to begin execution five days after the scenario begins, a "5" would appear in this field. Up to 15 plans may be considered in a scenario; one record is required for each plan.

PLANDAT - FIRST RECORD:

Field Description	Internal Format	Column Variable	Number
MAXIMUM DAYS LATE - UNITS	I3	MXLTUN	1-3
MAXIMUM DAYS LATE - REPLACEMENTS	I3	MXLTPX	4-6
MAXIMUM DAYS LATE - RESUPPLY AMMUNITION	I3	MXLTRA	7-9
MAXIMUM DAYS LATE - GENERAL RESUPPLY	I3	MXLTRO	10-12
MAXIMUM DAYS LATE - RESUPPLY POL	I3	MXLTRP	13-15
AFOE MIX PARAMETER	A1	AFOEMX	16
AMMO MIX PARAMETER ('Y'/'N')	A1	(MIX)	17
WAR OR PEACE ('W'/'P')	A1	(PW)	18
REFUELING DELAY (days)	F3.1	BUNKDE	19-21
LOAD DATE	I6		22-27
LOAD TIME	I4		28-31

PLANDAT - SUBSEQUENT RECORDS:

Field Description	Internal Format	Column Variable	Number
PLAN IDENTIFICATION	A12	SOPLAN	1-12
DAYS OFFSET	I4	OFFSET	13-16

VIEWDAT

Field Description	Internal Format	Column Variable	Number
BEGINNING C-DAY OF THIS PHASE	I3	PPTIME	1-3
FILLER			4
STATUS OF PANAMA CANAL ('O'/'C')	A1	(CANCAS)	5
FILLER			6
STATUS OF SUEZ CANAL ('O'/'C')	A1	(CANCAS)	7
FILLER			8
NAME OF RISK AND OCEAN AVOIDANCE FILE	A8	RNAME	9-16

PARADAT

This file usually contains one record for each parameter. (Exceptions will be noted). This design has permitted maximum flexibility in adding and deleting parameters during program development. The use of these parameters is described in Paragraph 5.3. Subsystems requiring each variable are noted in the field description.

Field Description	Variable Type	Internal Variable
Early interval widths, days--aggregation	Integer	INT1
C-day on which late interval widths begin--aggregation	Integer	DIVDAT
Late interval widths, days--aggregation	Integer	INT2
Weight given to empty space in computing a ship's score--heuristic routing	Real	PAR1
Tolerance on cost when finding best ship--heuristic routing	Integer	PAR2
Tolerance on cost above best ship for finding good ships--heuristic routing	Integer	PAR3
Days improvement required to refine a load--load refining process	Integer	PAR4
Maximum increase in route distance for selection of a candidate channel, miles--heuristic routing	Integer	PAR5
Days greater than prime available to load date (ALD) allowed for secondary ALD--heuristic routing	Integer	PAR6
Limit on fractional route distance increase for adding secondary channels--all routing	Real	PAR7
Secondary channel sail date must be no more than PAR8 days greater than prime sail date--heuristic routing	Integer	PAR8
Maximum distance between ports of embarkation or debarkation for clustering channels, miles--aggregation	Integer	PAR9
Time span for determining minimum port activity, days--aggregation	Integer	PAR10
Minimum load, in measurement tons, for a ship to dock--aggregation and simulation	Integer	PAR11
Minimum unload for a ship to dock--aggregation and simulation	Integer	PAR12
Cost of cargo on dummy ship--routing and loading	Integer	PAR13
Multiplier on minimum ship load to eliminate ships before the ship loading subsystem	Real	PAR14
Miles a ship will go to pick up evacuees (NEO) as prime cargo--all routing	Real	PAR15
Adding secondary channels to a ship depends on the number of alternatives the channel has; more than PAR16 is considered 'many' alternatives--all routing	Integer	PAR16
Similar to PAR16, more than PAR17 is considered 'some' alternatives--all routing	Integer	PAR17
Multiplier on the slopes of the acceptance criteria curves for secondary channels (see Fig. 5.1)--routing	Real	PAR18

PARADAT (continued)

Field Description	Variable Type	Internal Variable
When evaluating a secondary channel, the delivery of that channel by an alternate ship has PAR19 days added to account for interaction with other channels--all routing	Integer	PAR19
Cost plateau for unsailed ships--optimization-based routing	Integer	PAR20
Time, in days, to sheath a ship--simulation	Real	PAR21
Multiplier on maximum days of lateness; used in the cost function to price unfavorable channel-ship combinations out of consideration--all routing	Real	PAR22
Maximum iterations in the transportation problem	Integer	ITMXTP
Draft, in feet, of barges for Lash and Seabee ships	Integer	BARGE
True if port dimensions are not constraining	Logical	UNPORT
A nominal load time, in days--aggregation and routing	Real	LOAD
A nominal unload time, in days--aggregation and routing	Real	UNLOAD
Multiplier on distance to account for refueling time--all routing	Real	BUNKER
Time, in days, to dock a ship	Real	DOKTIM
Spare	Integer	SPARE1
Spare	Integer	SPARE2
Factor to convert square feet to measurement tons	Real	SQFTMT
Factor to convert square feet to container stowage	Real	SFCONT
Minimum fraction of ship load needed to sail ship type I during interval j:j=1 is first 30 days j=2 is next 30 days j=3 is after day 60; 20 records, 3 fields each--routing and simulation	Real	LDMINj(I)
Default value of ship fuel capacity, in long tons, for ship type I; 2 records, 10 fields each--input	Integer	DEFCAP(I)
Default value of ship fuel use rate, in long tons per day for ship type I; 2 records, 10 fields each--input	Integer	DEFUSE(I)
Minimum fraction of fuel load retained at all times--bunkering	Real	MNFUEL
Assumed initial fraction of fuel on board ship--bunkering	Real	INFOB
Maximum delay before stranding cargo at an intermediate port--simulation	Integer	WAIT
Available to load date will not exceed the ready to load date by more than RLDALD days--input	Integer	RLDALD
Maximum ship length, feet, Panama Canal	Integer	PANL
Maximum beam, feet, Panama Canal	Integer	PANB
Maximum draft, feet, Panama Canal	Integer	PAND
Maximum ship length, feet, Suez Canal	Integer	SUEZL
Maximum beam, feet, Suez Canal	Integer	SUEZB
Maximum draft, feet, Suez Canal	Integer	SUEZD
Canal transit time, days; 2 fields: Panama and Suez	Real	CTRAN

PARADAT (continued)

Field Description	Variable Type	Internal Variable
File name for saving distances	Character*7	DISSAV
File name for saving ship availability dates	Character*7	SADSAV
File name for reading distances	Character*7	DISGET
File name for reading ship availability dates	Character*7	SADGET
File for saving interport attrition risks	Character*7	ATTSAV
File for reading interport attrition risks	Character*7	ATTGET
Multiplier on estimated ship availability date--all routing	Real	SADMUL
Maximum days delay at intermediate port that assures the same ship will be used for both trip legs	Integer	IDMIN
File name for saving the ship routing solution	Character*7	SOLSAV
File name for reading the ship routing solution	Character*7	SOLGET
If true, cargo is "pushed" to the intermediate port, even if it cannot be delivered to its final port	Logical	PUSH
The random number seed, used in attrition	Integer	SEED
If TRUE, fuel bunkering will be calculated	Logical	DOBUNK
If TRUE, route refining will occur	Logical	DOREEF
Maximum days earliness allowed for a delivery	Integer	MAXEAR
If TRUE, the heuristic model is used	Logical	SWNOPT
If TRUE, heuristic model results will be used to select the ships to be available during the optimization-based routing	Logical	SWSEL
If TRUE, the standard optimization-based routing procedure will be used, otherwise an experimental version will be used	Logical	SWAB
If TRUE, the general network code will be used to load ships, otherwise the routing solution will be sent directly for simulation	Logical	SWGNC
If TRUE, Korean ships are limited to Korean ports of debarkation	Logical	KOREA
If TRUE, NATO ships are limited to NATO ports of debarkation	Logical	NATO
A large number which dominates the cost function	Integer	BIGCST
Time reflow begins (future use)	Integer	CNOW
The type of run being made: 'R': required lift 'L': lift capability 'C': closure analysis (normal SAIL run) 'A': external aggregation 'D': stand-alone distance 'S': stand-alone distance and ship availability dates	Character	RUNTYP

PARADAT (continued)

Field Description	Variable Type	Internal Variable
Cutoff date for lift capability calculations	Integer	LCCUT
Unit aggregation switch:	Character	SWUNIT
'A': only aggregate cargos with similar unit identifiers		
'B': aggregate on the basis of unit or non-unit cargo		
'C': do not use unit identifiers in aggregation		

VIEWDAT

This file controls the "view" of the world through time; the changing canal conditions, attrition risks, and ocean avoidance areas are defined.

The record format is

Field Description	Variable Type	Internal Variable	Column Number
Beginning C-day of condition	I3	PPTIME	1-3
SPACER	1X		4
Panama condition ("O" or "C")	A1	CANST	5
SPACER	1X		6
Suez condition ("O" or "C")	A1	CANST	7
SPACER	1X		8
RISKDAT file name (MT, CBBLS, or PAX per day)	A7	RNAME	9-15

TOTAL LENGTH OF ALL FIELDS = 15 CHARACTERS

THRUDAT

Port throughput may change over time; for those facilities that have changes, this file gives the date of the change and the new throughput rate.

The record format is

Field Description	Variable Type	Internal Variable	Column Number
GEOLOC code of a port with a change	A4		1-4
Facility type changed (A, B, C, G, P)	A1		5
C-day the change becomes effective	I3	TCAP	6-8
Throughput starting on that date	I7	MCAP	9-15

TOTAL LENGTH OF ALL FIELDS = 15 CHARACTERS

LCAPDAT

For lift capability computations, a file is needed to describe the "channels" to be carried by the available ships. Each channel describes a port of embarkation, a port of debarkation, the type of cargo carried over the channel, and the fraction of the total deployment represented by the channel. It is assumed that this file will be produced by a user interface; hence, the internal SAIL cargo code is used.

There is one record for each channel; the record format is

Field Description	Variable Type	Internal Variable	Column Number
GEOLOC code of port of embarkation	A4	CPOE	1-4
SPACER	A1		5
GEOLOC code of port of debarkation	A4	CPOD	6-9
Internal SAIL cargo code	I3	CTYPE	10-12
Fraction of cargo on the channel	F6.3	CQTY1	13-18

TOTAL LENGTH OF ALL FIELDS = 18 CHARACTERS

POXSDAT

For the special transfer port problem (see Paragraph 4.3.28), two additional data files are required; one listing the ships that are restricted from going to certain ports, and another that gives the transfer port and the ports from which the special ships are restricted. The POXSDAT file lists the restricted ships, one record for each ship. The records are five characters wide and contain the NISC number of the restricted ships.

POXPDAT

The first record of this file contains the GEOLOC code of the special transfer port; subsequent records contain the GEOLOC code of ports from which the ships listed in the POXSDAT file are restricted. The records are four characters in length.

5.2.3.3 Standard Reference Data Files

The files in this paragraph describe characteristics of ships, facilities, and geography which change infrequently. The information contained in the first four of these files is shown in Appendix E.

SHPTDAT

Each type of ship may have one or more types of stowage compartments. The potential types, and the integer index associated with each type, are: (1) breakbulk, (2) lift-on/lift-off square feet, (3) container stowage, (4) passengers, (5) petroleum, and (6) roll-on/roll-off square feet. The model uses the integers one to six to refer to these stowage types.

The first record for each type of ship in the SHPTDAT file contains an integer-valued ship type code, the alphabetic descriptor of the ship type, an alphabetic descriptor of the first compartment type to be described, the integer corresponding to that stowage type, an indication of whether or not the compartment is a primary compartment (or incidental, such as deck space), and finally, the nominal load/unload rate for cargo on this type of ship. (This last item is used in early stages of the scheduler to estimate loading and unloading times.) If there is more than one type of compartment for a ship, additional records are included with blanks in fields: ship type number, ship type description, and nominal load/unload rate.

The record format is

Field Description	Variable Type	Internal Variable	Column Number
SHIP TYPE NUMBER (see Section 5.5)	I2	SHPTNR	1-2
SPACER	1X		3
SHIP TYPE DESCRIPTION	A16		4-19
COMPARTMENT DESCRIPTION	A8		20-27
COMPARTMENT INTEGER INDEX	I1		28
SPACER	1X		29
PRIME COMPARTMENT INDICATOR ('Y'/'N')	A1	(SHPCAP)	30
NOMINAL LOAD/UNLOAD RATE (MT, CBBLS, or PAX per day)	I7	NLST	31-37

TOTAL LENGTH OF ALL FIELDS = 37 CHARACTERS

LOADDAT

The entries in the LOADDAT file are integers representing the load rate for a given cargo class in a given compartment on a given ship type, in measurement tons per day or hundreds of barrels per day. The first field in each record gives the ship type number and the second field gives the stowage index. The remaining fields are grouped so that a record defines all load rates for a particular ship-type-compartment. If there is more than one compartment on a ship type, additional records are included with blank ship type numbers. The format is the following:

Field Description	Variable Type	Internal Variable	Column Number
SHIP TYPE NUMBER (see Section 5.5)	I2		1-2
COMPARTMENT INTEGER INDEX	I2		3-4
LOAD RATES (MT, CBBLS, or PAX per day)	11I6	LDRATE	5-70

TOTAL LENGTH OF ALL FIELDS = 70 CHARACTERS

UNLDDAT

This structure of this file is identical to the LOADDAT file; it contains unload rates in measurement tons per day or hundreds of barrels per day.

STOWDAT

The entries in the STOWDAT file are real numbers representing the fractional stowage factor for a given cargo class in a given compartment on a given ship type. The first field in each record gives the ship type number, and the second field gives the stowage index. The remaining fields are grouped so that a record defines all stow factors for a particular ship type-compartment. A final field gives the compartment-related stow factor used (see below). If there is more than one compartment on a ship type, additional records are included with blank ship type numbers. There should be one record for each ship-compartment defined in the SHPTDAT file. The format is the following:

Field Description	Variable Type	Internal Variable	Column Number
SHIP TYPE NUMBER (see Section 5.5)	I2		1-2
COMPARTMENT INTEGER INDEX	I2		3-4
STOW FACTORS	11F6.2	STOW	5-70
COMPARTMENT-RELATED STOW FACTOR	F6.2	STOWST	71-76

TOTAL LENGTH OF ALL FIELDS = 76 CHARACTERS

One final record on the file gives the cargo class-related stowage factor which, in conjunction with the compartment-related factor, is used in estimating stow factors in the Ship Routing Subsystem (see Paragraph 4.2.5). This record is identified to the system by having "99" placed in the compartment integer index field. The data in this record are placed into the variable STOWCG.

FACIDAT

The file FACIDAT designates the types of port facilities required to load or unload each cargo class on each ship-compartment type. As many as two facilities can be made available for processing a given cargo class on a given ship compartment, the first being preferred over the second. Every ship type needs a record for each of the six potential stow types; that is, six records per ship type. Each record contains a facility pair for each of the eleven cargo classes; the FORTRAN format for each record is (11(3X,A1,1X,A1)). One of six codes can appear at the intersection of a cargo class and a ship compartment; the code designates the kind of facility required. The facility codes are the following: "A" for ammunition, "B" for container ammunition, "C" for containers, "G" for general, "P" for petroleum, and "X" indicates the cargo class cannot be loaded at any facility.

The last record contains estimates for nominal processing times for nominal ships at each facility type, in tons per day (5I6).

COMPDAT

This file describes the various types of compartments found on ships, maps these to a cargo measure, and tells whether the compartment is used in dry cargo scheduling or petroleum scheduling. This is described further in Paragraph 5.6.

Field Description	Variable Type	Internal Variable	Column Number
COMPARTMENT DESCRIPTION			1-12
INTERNAL COMPARTMENT CODE			13-14
ABBREVIATED UNITS			16-21
INTERNAL STOW-TYPE CODE	I2	COMPST	22-23
COMPARTMENT UNITS			26-45
PROBLEM TYPE	I1	PRBTYP	62

TOTAL LENGTH OF ALL FIELDS = 62 CHARACTERS

PREFDAT

The various types of cargo may be in more than one type of ship, and often can be accommodated in more than one compartment type on a ship. The stowage generally varies in efficiency among the allowable compartments. This file defines the permissible cargo/ship/compartment combinations and indicates the preferences among them. There is a set of six records for each ship type, one for each of the six potential compartment types. Each record contains eleven fields, one for each of the cargo types. The fields are four characters in width and contain a right-adjusted integer, one through four. A "1" means the cargo/ship/compartment is a preferred combination, "2" and "3" are decreasing preferences, and a "4" indicates that either the ship-type does not have that type of compartment or that the cargo cannot be placed in it.

LINEDAT

The concept of "lines" being used in the distance computations was described in Paragraph 4.2.3. The LINEDAT file contains the endpoints and computed parameters for those lines. There is one record for each line that has been defined:

Field Description	Variable Type	Internal Variable	Column Number
LATITUDE OF POINT 1 ON LINE	F7.2	POINT	1-7
LONGITUDE OF POINT 1 ON LINE	F7.2	POINT	8-14
LATITUDE OF POINT 2 ON LINE	F7.2	POINT	15-21
LONGITUDE OF POINT 2 ON LINE	F7.2	POINT	22-28
LINE SLOPE	F10.4	COEF	29-38
LINE INTERCEPT	F10.4	COEF	39-48

TOTAL LENGTH OF ALL FIELDS = 48 CHARACTERS

NODEDAT

The nodes used in the distance computations are defined on this file, one record for each node (see Paragraph 4.2.3):

Field Description	Variable Type	Internal Variable	Column Number
NODE IDENTIFICATION	A12	NODEID	1-12
LATITUDE	F10.4	NLAT	13-22
LONGITUDE	F10.4	NLONG	23-32

TOTAL LENGTH OF ALL FIELDS = 32 CHARACTERS

LINKDAT

The links between the nodes are defined on this file, one record for each link. Note that the last field, the distance between the nodes, can be missing; if it is, it will be computed using a great circle distance measure. This computed distance will then be written into the file (see Paragraph 4.2.3):

Field Description	Variable Type	Internal Variable	Column Number
LINK IDENTIFICATION	A8	LINKID	1-8
NODE AT ONE END OF THE LINK	A12	NID1	9-20
NODE AT THE OTHER END	A12	NID2	21-32
DISTANCE BETWEEN THE NODES (miles)	I6	MILES	33-38
FIRST NODE NUMBERS	I6	(NODE1)	39-44
SECOND NODE NUMBERS	I6	(NODE2)	45-50

TOTAL LENGTH OF ALL FIELDS = 38 CHARACTERS

RISKDAT

The VIEWDAT file contains information about the state of the world beginning at a particular C-day; the state includes the status of the two major canals and the name of a file which describes the attrition risks and ocean avoidance areas beginning with the given C-day. RISKDAT is the generic name for those files. For example, suppose the VIEWDAT file contained the following two records:

```
 0  O  C  RISKDAT1
20  C  C  RISKDAT2
```

This would indicate that, beginning on C-day zero, the Panama Canal would be open, Suez would be closed, and the risk and ocean avoidance would be described by the file RISKDAT1. At day 20, the Panama Canal would close, and the risk and avoidance areas would be described by the file RISKDAT2.

Each RISKDAT-type file contains a digital map of the world. There is one data point for each latitude-longitude combination for all longitudes and for latitudes from 70 degrees South to 80 degrees North. Each data entry contains the probability that a ship at (or near) the given position will be lost for each day of exposure. Ocean avoidance areas are indicated by a negative number.

The RISKDAT files are binary; the first entry is for latitude 70 South, longitude 179 West. Subsequent entries fill in all the latitudes up to 80 North for this longitude. The next entry is for 70 South, 178 West...and so on until 80 North, 180 East is reached.

It is expected that these files will be automatically created by an interactive geographic program. Lacking that, a stand-alone FORTRAN program called INRISK is available to prepare the files.

5.2.4 Descriptions of Output Files

The output files divide into two main groups: those intended for internal system use or for the primary benefit of the system development, and those intended as the major interface files with the SEASTRAT system.

5.2.4.1 Internal System and Development Files

CHANOUT

CHANOUT contains a report of the results of CHANEL, the cargo aggregation routine. In this file, numbers are used to represent port names and program variable names head some of the columns; this report was designed only for program development purposes.

INTPORT

This file contains the interport distances for all possible canal conditions (all distances are in miles).

Field Description	Variable Type	Internal Variable	Column Number
FIRST PORT GEOLOC CODE	A4	PID	1-4
SECOND PORT GEOLOC CODE	A4	PID	5-8
DISTANCE WITH BOTH CANALS OPEN	I5	D	9-13
CANALS USED, IF ANY	A1	C	14
DISTANCE WITH BOTH CANALS CLOSED	I5	D	15-19
ALWAYS BLANK	A1	C	20
DISTANCE WITH PANAMA CANAL OPEN	I5	D	21-25
CANALS USED, IF ANY	A1	C	26
DISTANCE WITH SUEZ CANAL OPEN	I5	D	27-31
CANALS USED, IF ANY	A1	C	32

TOTAL LENGTH OF ALL FIELDS = 32 CHARACTERS

LISTOUT

As the Simulation Subsystem executes, the loading and unloading events are recorded on the LISTOUT file. At the end of the simulation, this file contains a complete record of the relevent activities: how much of each cargo was loaded on each ship, when the load and unload occurred, and which ship compartment was used. The file is then read by the Reporting Subsystem and the information merged with other data to build the major output file, FINIOUT.

If the optimal ship loading option is selected, a LISTOUT file is written by the simulator to permit the Loading Subsystem to obtain timing information for the ships (see Paragraph 4.2.6).

The LISTOUT file is a binary file; such files typically can be written and read in considerably less time than ASCII formatted files. Unlike the other file descriptions, the Variable Type column for this file gives the true FORTRAN type rather than the format used to write the variable.

Field Description	Variable Type	Internal Variable
SEQUENCE NUMBER OF CARGO ON SHIP	Integer*2	SEQOS
TIME OF SHIP'S ARRIVAL IN PORT	Real	TARRIV
TIME ACTIVITY BEGINS	Real	TNOW
TIME ACTIVITY ENDS	Real	TTHEN
SHIP NISC NUMBER	Integer*4	SID
INTERNAL PORT NUMBER	Integer*4	P
ACTIVITY CODE (load, unload, etc.)	Character*1	CODE
AMOUNT OF CARGO (units as in STOW TYPE)	Integer*4	CSAMT
DAYS LATE (unload records only)	Integer*4	LATE
POINTER FOR SPLIT CARGOS	Integer*4	IPOINT
STOW TYPE	Integer*4	STP
INTERNAL SHIP NUMBER	Integer*4	S
INTERNAL TRIP NUMBER	Integer*4	TR
INTERNAL CARGO NUMBER	Integer*4	C
LATEST ARRIVAL DATE	Integer*2	LAD
PORT FACILITY TYPE	Integer*2	FACAT
CHANNEL NUMBER	Integer*4	CH
RISK OF ATTRITION	Real	RSKOS
ACCUMULATED PORT QUEUING TIME OF THE SHIP	Real	QSHIP

TOTAL LENGTH OF ALL VARIABLES = 67 BYTES

EXCLOUT

This file contains a list of cargo excluded from the schedule for any of several scheduling problems as described in Paragraph 5.6. Originally intended as a SEASTRAT interface file, the information on EXCLOUT has now been appended to the FINIOUT file:

Field Description	Variable Type	Internal Variable	Column Number
EXCLUSION CODE (see Section 5.6)	I1	CODE	1
AMOUNT OF EXCLUDED CARGO	I7	AMOUNT	2-8
STOW TYPE	I3	STP	9-11
SCHEDULED TIME, IF KNOWN	F9.2	TIME	12-20
INTERNAL CARGO NUMBER	I6	C	21-26

TOTAL LENGTH OF ALL FIELDS = 26 CHARACTERS

TESTOUT

TESTOUT is primarily used to store temporary testing or debugging information.

EXEPOUT

EXEPOUT is the exception report that displays some of the options selected and reports problems encountered during a run.

SPECDAT

This file is only created by the special transfer port version of SAIL; as was the LISTOUT file, it is an intermediate file of no concern to the system interface. It contains information about the cargos that are placed on special ships in the first pass through the algorithm in order to assure these cargos are treated similarly on the second pass (see Paragraph 4.3.28 for a description of this process). For each cargo carried on a restricted ship, a record is written that contains the internal SAIL cargo number, the short tons, measurement tons, square feet, numbers of passengers, or hundreds of barrels of POL carried, and the special ship's NISC number. The seven fields are each ten characters wide.

5.2.4.2 Major System Interface Files

HISTOUT

HISTOUT contains summary reports used to evaluate runs. Information includes closure date, total mileage, "measure of goodness," delivery histograms by cargo class, summary cargo table, summary ship table, summary port table, ship type, utilization summary, and, if selected, bunkering needs. Specialized information is displayed for required lift analyses.

DISTOUT

DISTOUT is a chronological list of all shipping legs. For each leg there is a record containing the following:

Field Description	Variable Type	Internal Variable	Column Number
NISC NUMBER	I5	SID	13-17
SPACE	1X		
PORT GEOLOC CODE	A4	PID	19-22
SPACE	1X		
DATE LOAD BEGINS OR UNLOAD ENDS	F7.2	TNOW	24-30
SPACE	1X		
NEXT PORT TO VISIT	A4	PID	32-35
SPACE	1X		
ARRIVAL DAY AT NEXT PORT	I3	TNEXT	37-39
SPACE	1X		
DISTANCE TO NEXT PORT (miles)	I5	D	41-45
BUNKERING NEEDS (' Y'/' ')	A2	BNKR	46-47
ATTRITION DAY	I4	ATTRIT	48-51
CANALS USED: 0=NONE, 1=PANAMA, 2=SUEZ, 3=BOTH	I2	CANALS	52-53
BUNKER DIVERSION INDICATOR: 0=NO DIVERSION OR DIVERSION SUCCESSFUL 1=UNSUCCESSFUL DIVERSION	I2	DIW	54-55
SPACE	1X		56
NATURE OF ACTIVITY AT FIRST PORT	A1	ACSEQ	57
TRIP NUMBER	I3	TRSEQ	58-60

TOTAL LENGTH OF ALL FIELDS = 60 CHARACTERS

FINIOUT

This is the primary output file. It gives a complete itinerary of each ship, including all cargo carried, ports visited, and times of operations. The file structure is the following:

Field Description	Variable Type	Internal Variable	Column Number
OPLAN	A6	SNARIO	1-6
CYCLE	A2	SNARIO	7-8
RUN	I3	SNARIO	9-11
MODULE	A1	SNARIO	12
REQUIREMENT IDENTIFICATION	I6	RID	13-18
UNIQUE IDENTIFYING NUMBER	I2	ISEQ	19-20
ACTIVITY CODE 'L' = LOAD AT POE 'U' = UNLOAD AT POD 'M' = LOAD AT POI 'I' = UNLOAD AT POI 'E' = EXCLUDED RECORD	A1	IACT	21
STOW TYPE	I1	ISTYPE	22
SHORT TONS	F8.1	(STONS)	23-30
MEASUREMENT TONS OF CARGO	F8.1	AMT1	31-38

FINIOUT (Continued)

Field Description	Variable Type	Internal Variable	Column Number
SQUARE FEET OF CARGO	F8.1	AMT2	39-46
NUMBER OF PASSENGERS	I6	IPAX	47-52
AMOUNT OF PETROLEUM (CBBLS)	I6	IPOL	53-58
STOW ADJUSTED AMOUNT OF CARGO	F8.1	AMT3	59-66
FACILITY GEOLOCATION CODE	A4	PID	67-70
COMPLEX GEOLOCATION CODE	A5	COMPLX	71-75
TYPE OF PORT FACILITY	A1	FAC	76
SHIP TYPE	I2	SHPTNR	77-78
SHIP IDENTIFICATION	I5	SID	79-83
TRIP NUMBER	I2	ITR	84-85
ARRIVAL DATE OF SHIP IN PORT	I4	TARR	86-89
SCHEDULED DAY FOR ACTIVITY TO BEGIN	I4	DATE2	90-93
EARLIEST DAY ACTIVITY CAN BEGIN	I4	DATE	94-97
LATEST DAY FOR ACTIVITY TO BEGIN	I4	DATE3	98-101
CARGO DELIVERY DATE	I4	DEPDEL	102-105
LATENESS (EARLINESS)	I4	ILATE2	106-109
DEPARTURE DATE FROM PORT	I4	DEPOTE	110-113
CODE FOR NON-DELIVERY	I1	ERCODE	114
ORIGIN READY TO LOAD DATE	I3	RLD	115-117
POE AVAILABLE TO LOAD DATE	I3	ALD	118-120
INTERMEDIATE DAYS DELAY	I3	DDELAY	121-123
POD EARLIEST ARRIVAL DATE	I3	EAD	124-126
POD LATEST ARRIVAL DATE	I3	LAD	127-129
FUEL TYPE	A3	FUELTP	130-132
REQUIREMENT CONSTRAINTS	A1	REQCON	133
UNIT	A6	UNIT	134-139
RISK TO WHICH CARGO IS EXPOSED	F6.4	RSKOS	140-145
CARGO TYPE	I2	CTYPE	146-147
ACCUMULATED SHIP QUEUE TIME	F6.1	QSHIP	148-153
REMAINING CAPACITY	I6	CAPREM	154-159

TOTAL LENGTH OF ALL FIELDS = 159 CHARACTERS

5.2.4.3 Files for Plotting Routes

To accommodate the plotting of ship routes or the routes between ports, two special types of output files are generated. The route file, called ROUTE.RTE, is a binary file containing sequences of locations. Each record contains a latitude/longitude pair defining a specific spot; a series of these records defines a route. (The lat/lons are signed real numbers with the convention that western longitudes and southern latitudes are negative.)

The other type of file is the pointer file. Basically, each pointer file gives the beginning and ending record number of the route file to be used to define a particular route. For example, the port-to-port pointer files contain the geolocation codes of the two ports and the first and last records of the route file to be used in plotting the route between the ports.

There are five pointer files. The first four give the routes between pairs of ports with various canal conditions in effect: ALLOPEN.PTR when both canals are open,

PANSHUT.PTR for the Panama Canal closed, SUESHUT.PTR for the Suez Canal closed, and ALLSHUT.PTR for both canals closed. The format of each record is as follows:

Field Description	Variable Type	Internal Variable	Column Number
GEOLOC code of the first port	A4		1-4
GEOLOC code of the second port	A4		5-8
First record of the ROUTE.RTE file	I6		9-14
Last record of the route file	I6		15-20

TOTAL LENGTH OF ALL FIELDS = 20 CHARACTERS

The final file is the ship pointer file, SHIPS.PTR. It contains pointers to route records for a ship's voyage between two ports on each record; an alternate set of pointers gives the route that would result from a change in canal status at a given date during the ship's voyage. The record format is as follows:

Field Description	Variable Type	Internal Variable	Column Number
Ship's NISC number	I5		1-5
First route record, original voyage	I6		6-11
Last route record, original voyage	I6		12-17
First route record, altered voyage	I6		18-23
Last route record, altered voyage	I6		24-29

TOTAL LENGTH OF ALL FIELDS = 29 CHARACTERS

5.3 PARAMETERS AND OPTIONS

A variety of parameters and input options control the features and operation of the SAIL model. Some of these parameters, such as the assumed status of a canal, are under the planning analyst's control each time a planning case is run. Other parameters are considered to be matters of the Military Sealift Command's policy and would be infrequently changed. A final type of parameter controls technical features of the program's operation and should only be changed by the software maintenance staff. It is not the purpose of this report to assign the parameters to these categories. The purpose of this section is to explain each parameter and to describe its impact, allowable value range, and location in the files.

5.3.1 Time Division and Aggregation Parameters

The time period of the plan is divided into intervals for the purpose of cargo aggregation (see Paragraph 5.1.3.3). It may be desirable to have only a few days in each interval at the beginning of the plan, and longer intervals later in the plan. The parameters in file PARADAT are as follows:

INT1	Initial interval width, in C-days
DIVDAT	Date of change to longer intervals
INT2	Interval width after the date of change

Tests suggest that a reasonable range for interval sizes is between 5 and 20 days; even shorter intervals may be needed for special situations, but general use of shorter intervals is not advised. Longer intervals result in increased aggregation of cargo, which reduces the size of the ship routing problem (and hence decreases processing time). More aggregation can also lead to more efficient loading of ships. On the other hand, longer intervals decrease the precision of the consideration of lateness, which the next paragraph will show to be a key ingredient of the cost computations.

Other parameters affecting aggregation involve the minimum amount of cargo that will be loaded or unloaded during a port call by a ship. The minimum cargo load, in measurement tons, is PAR11 and the minimum cargo unload, also in measurement tons, is PAR12. These two parameters are also used in the Simulation Subsystem. During aggregation, however, the number of ships involved in cargo movements at a port is not known; if the total expected activity at a port during some period of time is insufficient, as defined by PAR11 or PAR12, the cargos can be eliminated because no ship will stop for them. The period of time, in days, is defined by PAR10.

Nominal values and ranges are as follows:

Parameter	Nominal Value	Possible Range
PAR10	5	0 to 20
PAR11	200	0 to 5000
PAR12	200	0 to 5000

5.3.2 Cost Coefficients (or Penalties)

The transportation problem formulation was described in Paragraphs 4.2.5 and 5.1.5.15. One requirement of that formulation was the development of a penalty cost representing the desirability of assigning a particular aggregation of cargo (called a channel) to a compartment of a given trip of a ship. Such a cost is found for all cells of the transportation problem which represent feasible cargo assignments.

The following combinations are considered infeasibilities and are therefore assigned very high costs in order to prevent their access to the solution.

> Some ships are not compatible with certain ports. Some of these mismatches are maintained by consistent use of the load/unload/stow/facility tables in Appendix E. In addition, if parameter UNPORT is false, the ship dimensions must fall within the dimensional limits of the port facility.

Breakbulk ships will not be routed based on deck space assignments except for oversized cargo or floating craft. (This restriction is relaxed in the Ship Loading Subsystem.)

Competition with preferred ship assignments is eliminated in a similar manner.

And finally, projected lateness exceeding the product of the parameters PAR22 and MXLTxx will be priced out of the solution.

After eliminating the above mismatches, a cost function is invoked which will produce several cost tiers, each of which dominates all lower tiers. The first tier (highest cost) penalizes lateness. Late penalties are now in three categories: 75% of maximum allowable lateness; more than five days late; and less than five days late. These penalties are listed from highest cost to lowest and *any* late penalty will dominate *all* other cost criteria.

The next cost tier encourages the use of ships already in the schedule, in order to minimize shipping resources.

Next, the use of favored compartments is encouraged. Currently, a favored compartment is one whose stowage type (MT, SQFT, etc.) agrees with the stowage type of the candidate channel.

And finally, the ship proximity to the POE--measured by ship availabilities--encourages the use of nearby ships.

The cost for assignment of any cargo to the dummy ship is a constant equal to PAR13. Cargo placed on the dummy ship, it may be recalled, is the equivalent to not moving the cargo. PAR13 is higher than any cost associated with a legitimate assignment.

There is an additional cost plateau used in the optimization version of the ship routing which penalizes all ships equally--until a ship is selected for routing. Then the plateau is removed for the selected ship and the problem reoptimized. The effect is to ensure that the ship under consideration is filled (insofar as is possible) with prospective cargos. The level of this plateau is PAR20.

Schedule earliness may be limited through the use of parameter MAXEAR, which specifies the maximum days early allowed for a delivery.

Nominal values of the various cost parameters are now given; at the time of publication, only limited experiments with these parameters had been conducted. With this caveat, guidance about reasonable ranges is given.

Parameter	Nominal Value	Possible Range
PAR13	90000	Significantly higher than any legitimate cost calculated; should be about twice PAR20
PAR20	40000	Higher than any legitimate cost
PAR22	1.10	1.00 to 2.0
MAXEAR	30	0 to 999

All cost parameters are in the file, PARADAT.

5.3.3 Ship Routing Parameters

Before undertaking a discussion of the parameters related to ship routing, let us review the routing process, which involves routing one ship trip at a time. The steps taken to construct a ship trip are as follows:

1. set up an appropriate transportation model and solve it,
2. find a dominant cargo assignment on a ship,
3. remove incompatible cargo from consideration for this trip,
4. adjust costs and re-optimize the model, and
5. add compatible secondary channels from the solution vector until the list is exhausted or the ship is adequately loaded.

This process uses several parameters, which are defined as follows:

1. Secondary channels are rejected if they cause the ship to extend its route too much. If the trip distance with the secondary channel exceeds PAR7 times the distance with the primary channel alone, the secondary is considered incompatible.

2. If the insertion of a secondary channel has undesirable consequences, such as additional lateness, that channel will usually not be rejected until it has been determined that other, perhaps better, alternatives exist. Basically, the more alternatives that are available, the less likely the process is to accept the ill effects of the addition of this candidate channel. Three regions are defined by parameters; these set limits on the number of alternatives needed to characterize a channel as having "some" alternatives, and therefore a better chance of acceptance, or having "many" alternatives and less apt to be accepted. PAR17 is the minimum number of alternatives to be categorized as "some" and PAR16 defines "many" alternatives.

3. The curve used to accept or reject a channel based on alternatives is linear and the slope of the curve with less than "some" alternatives is PAR18; with "some" is 2 times PAR18; and with "many" is 5 times PAR18.

4. Alternative delivery dates are estimated as if the alternative ship would carry only the candidate channel and PAR19 will be added to account for multiple port calls.

5. A ship compartment must be sheathed before ammunition can be accommodated; the first ammunition loaded in any ship compartment must be delayed accordingly. PAR21 represents this sheathing delay in days.

6. Ships that fail to be loaded to PAR14 times minimum ship load are dropped from the routing prior to the final ship loading.

7. Cargos are clustered together based on less stringent criteria than required for placing them in similar channels. Their ports of embarkation and debarkation are not necessarily the same, but must be within PAR9 miles of each other.

8. It is generally intended that ships pick up evacuees (NEO) only on route to ports of embarkation, either initially or on return trips. The number of miles a ship will travel to pick up NEO is therefore limited--the limit is PAR15 miles.

In addition, the heuristic routing option uses the following parameters:

9. The best ship to carry the prime cargo is selected based on the cost matrix and other criteria. PAR2 defines the variance from the least cost permitted when evaluating those other criteria. Ships other than the best ship are also considered to be candidates for carrying the prime channel. PAR3 is the tolerance above the best ship's cost used to select the candidates.

10. Ships are evaluated for their efficiency in carrying the prime channel and those channels clustered with it. PAR1 is the weight given to empty space left on the ship in the evaluation formula. Higher values of PAR1 will therefore increase the penalty for selecting a ship larger than apparently necessary.

11. When adding channels to a ship, the new channel can increase the ship's route by, at most, PAR5 miles. Also, the new channel's estimated sailing date (CHT1) can be no more than PAR8 days beyond the prime channel's sailing date and the channel's available to load date (CHT0) can be no more than PAR6 days greater than the prime's.

Nominal values and ranges are

Parameter	Nominal Value	Possible Range
PAR1	0.5	0 to 1.0
PAR2	2	0 to 1000 (depends on cost function)
PAR3	5	0 to 1000 (depends on cost function)
PAR5	75	0 to 25000
PAR6	5	0 to 50
PAR7	1.4	1.0 to 2.0
PAR8	20	0 to 50
PAR9	100	0 to 1000
PAR14	0.5	0.2 to 1.0
PAR15	500	0 to 25000
PAR16	10	10 to 50
PAR17	10	5 to 20
PAR18	1.0	1.0 to 10.0
PAR19	5	3 to 8
PAR21	0.4	0 to 1.0

All of the ship routing parameters are in the file, PARADAT.

5.3.4 Ship Operating Parameters–Initial Estimates

At the beginning of a SAIL run, nominal values must be used for some ship operations since the actual ship to be used is not yet known. Other factors in the time required for a ship to make a trip are not explicitly known from other data. The following parameters, contained in the PARADAT file, fill some of the voids.

LOAD is a nominal time (in days) for loading a cargo on a ship.

UNLOAD is a nominal unload time.

BUNKER is a multiplier on travel times to provide time for bunkering.

DOKTIM is the nominal time to dock a ship. Currently, all ships require the same amount of time to dock and this number is used in all parts of the model.

SQFTMT converts square foot measures of cargo to measurement tons. This is used only when cargo data contain one measure but not the other.

SFCONT is the conversion factor from square foot measure to measurement tons of containers and is constant throughout the model.

SADMUL is a contingency multiplier used in the initial estimates of ship routes.

BARGE is the draft of barges associated with Lash and Seabee ships; the draft of these ships is set to this amount for port entry purposes.

DEFCAP is the default fuel capacity, in long tons, for each ship type.

DEFUSE is the default fuel consumption, in long tons per day, for each ship type.

Nominal values of these parameters are:

Parameter	Nominal Value	Possible Range
LOAD	0.2	0 to 2.0
UNLOAD	0.2	0 to 2.0
BUNKER	1.02	1.0 to 1.2
DOKTIM	0.2	0 to 1.0
SQFTMT	0.2	0.1 to 0.4
SFCONT	0.35778	none
SADMUL	1.0	0.1 to 3.0
BARGE	9	any integer
DEFCAP	Data for these two parameters must be supplied	
DEFUSE	by the Military Sealift Command.	

Additional ship parameters are found in the PLANDAT file; one parameter allows the analyst to specify that an attrition rate be applied to all active ships beginning on a particular C- day. Five sets of such dates and rates can be given, allowing attrition to change throughout the plan. Another parameter indicates whether or not ammunition can be placed on the same ship with non-ammunition. A parameter in the PLANDAT file also defines the "war" or "peace" status of the plan; this parameter impacts the initial availability of ships at ports of embarkation (see Paragraph 4.2.4) Finally, a refueling delay, in fractions of days, is specified.

5.3.5 Model Features Control Parameters

The following parameters in the PARADAT file control various features of the model.

5.3.5.1 Canal Restrictions

In order to gain access to a canal, a ship must be smaller than the canal dimensions. If a ship uses a canal, the appropriate canal transit time will be included in the trip transit time. These dimensions are on the PARADAT file.

PANL	Maximum ship length, Panama Canal: 950 feet
PANB	Maximum ship beam, Panama Canal: 106 feet
PAND	Maximum ship draft, Panama Canal: 39 feet
SUEZL	Maximum ship length, Suez Canal: 10000 (no limit)
SUEZB	Maximum ship beam, Suez Canal: 210 feet
SUEZD	Maximum ship draft, Suez Canal: 41 feet
CTRAN	Transit time through the canals, 1 day through each (this is a subscripted variable so the transit time for each canal can be given separately; fractions of days can be used)

In addition, the PLANDAT file may define opening and closing dates for each canal (see Paragraph 5.1.4.1). A C-day is given, followed by a character indicating the status of

each canal, "O" for open and "C" for closed. Four sets of dates and statuses can be given for a plan, allowing the canals to open and close several times.

5.3.5.2 Port Dimensional Constraints

UNPORT is a logical variable controlling whether port dimensional restrictions are to be used ("False" will apply restrictions).

Parameter	Nominal Value	Possible Range
UNPORT	TRUE	TRUE or FALSE

5.3.5.3 Gross Control of the Available to Load Date (ALD)

The time between the ready-to-load date at the origin (RLD) and the available-to-load date at the port of embarkation (ALD) can be controlled by the variable RLDALD. The ALD will be adjusted as follows: ALD = min(ALD,RLD+RLDALD). A very high value for RLDALD will cause the input ALDs to be used.

Parameter	Nominal Value	Possible Range
RLDALD	500	5 or greater; a large value leaves the available-to-load date unaltered

5.3.5.4 Delays at Intermediate Ports

If the dependent (second) leg of a trip with an intermediate stop is scheduled to leave before the independent (first) leg is completed, the simulation routine will delay departure of the dependent for up to WAIT days before stranding the cargo at the intermediate port.

The IDMIN variable allows the planner to use the same ship for both legs of a trip with an intermediate stop. Any cargo with an intermediate stop of less than IDMIN days will be kept on the same ship for both legs of the trip.

Typically, cargo with an intermediate stop will not be sent to its intermediate port if it cannot be delivered to its port of debarkation within the acceptable lateness criterion. When the logical variable PUSH is true, the cargo will be scheduled into the intermediate port even if the final leg would be delivered too late (and therefore eliminated from the problem).

Parameter	Nominal Value	Possible Range
WAIT	5	0 to 20
IDMIN	2	0 to 5
PUSH	FALSE	TRUE or FALSE

5.3.5.5 Pseudo-Random Number Generation Seed

The ship attrition feature uses random numbers which must be started with an integer variable SEED. Any arbitrary integer can be used to begin a random sequence.

Parameter	Nominal Value	Possible Range
SEED	0	any integer

5.3.5.6 Control of Underutilized Shipping Resources

SAIL is capable of eliminating underutilized ships between the resource allocation and the simulation. Minimum ship loads are specified by ship type and consist of three limits for each ship type in the variable LDMIN1-3: the minimum fraction of ship capacity allowed during the first 30 days of the plan; the minimum load during the second 30 days of the plan; and the minimum load thereafter.

Parameter	Nominal Value	Possible Range
LDMIN1	0.1	0.01 to 0.75
LDMIN2	0.1	0.01 to 0.75
LDMIN3	0.1	0.01 to 0.75

5.3.5.7 Route Refinement

The REFINE routines use one parameter; the number of days improvement in late deliveries must be at least PAR4 before a cargo will be moved from one ship to another. Refinement is invoked by setting parameter DOREEF to TRUE.

Parameter	Nominal Value	Possible Range
PAR4	10	0 to 20
DOREEF	TRUE	to refine routes

5.3.5.8 Bunkering

The bunkering computations require the assumed amount of fuel on board each ship at the beginning of the plan; this is defined for all ships by a fraction in parameter INFOB. The minimum fraction of fuel load to be maintained is described by the parameter MNFUEL. Default fuel capacities and fuel use rates, by ship type, are also in the PARADAT file, and are placed in the arrays FUELCP and FUELUS when these fields have been omitted on the SHIPDAT file. Bunkering computations and routings are invoked by setting parameter DOBUNK to TRUE.

Parameter	Nominal Value	Possible Range
INFOB	0.50	0.0 to 1.0
MNFUEL	0.3	0.1 to 0.4
DOBUNK	TRUE	to perform bunkering computations

5.3.5.9 Restricted Use of the Korean and NATO Fleets

Restricted use of Korean and NATO ships will be specified by setting parameters KOREA and NATO to TRUE, respectively. (This feature will not be functional until a country code is placed in the PORTDAT file and input records are designed to pass the needed information.)

Parameter	Nominal Value
KOREA	TRUE enables Korean fleet restrictions
NATO	TRUE enables NATO fleet restrictions

5.3.5.10 Control of Optimization Procedure

ITMXTP is the maximum number of iterations allowed to solve the transportation problem.

Parameter	Nominal Value	Possible Range
ITMXTP	1600	1000 to 5000 (on very large problems, this number may need to be increased)

5.3.5.11 Control of Solution Strategy

The variables used to select the solution strategy are described below.

SWNOPT	TRUE enables method one (NOPT)
SWSEL	TRUE will cause shipping resources selected by method one (NOPT) to be input to either method two (OPTA) or method three (OPTB)
SWAB	TRUE enables method two (OPTA), FALSE enables method three (OPTB)
SWGNC	TRUE enables the General Network solution of the Ship Loading Problem
CNOW	The start of the plan or set of plans; may be negative

RUNTYP Controls the type of analysis to be made:

- 'A': aggregate cargos only
- 'C': normal closure run
- 'D': distance calculations only
- 'F': find location of ships from origin, destination, and time
- 'L': lift capability computation
- 'P': project ships forward from their present location
- 'R': required lift computations
- 'S': ship availability date and distances only
- 'X': transfer port computations
- 'V': reuse deleted ships (see Paragraph 4.3.28)

LCCUT Cut off date for lift capability computations (meaningful only when RUNTYP = 'L')

SWUNIT Determines how cargo aggregation is to be performed on Unit Identification field of cargo records:

- 'A': aggregate only on similar unit descriptors
- 'B': aggregate on unit or non-unit
- 'C': no aggregation discrimination due to unit field

5.3.6 File Names to Save Intermediate Model Results

When a series of test cases are to be run, many of the calculations are the same from run to run. Recalculation can be avoided by saving intermediate results. The variables described here can be used to accomplish this goal. As long as the NODEDAT and LINKDAT files used to determine ocean distances, and the ports in a plan are unchanged, there is no need to recalculate the distance files. DISSAV and DISGET are the variables used to determine whether these calculations will be made or the distances will be read from an existing file. At least one of these variables should contain blanks. If distances are to be calculated and saved for use in subsequent runs, a seven-character file name should appear in variable DISSAV. Subsequent runs can access that precalculated distance file by inserting blanks in DISSAV and the seven-character name of the distance file in DISGET. Similarly, if the initial ship locations are unchanged between runs, SADSAV and SADGET can be used to avoid recomputation.

Examples:

If SADSAV = 'filenam' and SADGET= ' ', the ship availability dates will be saved in the specified file.

If SADSAV= ' ' and SADGET= 'filenam', an existing file of ship availability dates will be read.

SADSAV= ' ' and SADGET= ' ', ship availability dates will be calculated and not saved.

Finally, the results of a completed ship routing problem can be saved, and variations of the simulation model can be tested with the same ship routing as a beginning point. (This feature is primarily intended to be used in model development.) SOLSAV and SOLGET are the variables that would be set and used as described above.

5.4 PROGRAM DIMENSIONS

Sealift scheduling has been defined as a large, combinatorial problem. As such, it is not surprising that the SAIL formulation of that problem requires extensive computing resources. In this section, the memory requirements are considered. The FORTRAN language requires arrays and vectors to be "dimensioned" to the largest size expected in the problem. It is therefore a temptation to give arrays very large dimensions that will satisfy the demands of even the largest problems. The cost in memory requirements is significant, however. Some moderation and judgement will be required to keep the dimensions adequately sized, but under control. The main purpose of this section is to give guidelines on what "adequately sized" means for a particular planning scenario.

Changing the dimensions of a large set of programs is a demanding task. In the SAIL model, several dimensions are related; changing one dimension implies a change of other dimensions. This modification must be accomplished uniformly throughout the program. The SAIL source code has been organized to aid in this process. All software required for redimensioning is contained in common blocks which have the file extension .CMN.

An important subroutine, called INIDIM, is associated with the commons. This subroutine is used to initialize the maximum dimension size of all arrays. For example, the variable SDIM is used to specify the maximum number of ships that can be considered by the model. Suppose that number is 700. Subroutine INIDIM would then contain a PARAMETER statement that sets the number into the variable:

```
PARAMETER (SDIM = 700)
```

If the maximum number of ships changes to 750, this statement would need to be changed accordingly.

The dimensions in the COMMON blocks are then able to use the variable SDIM. For example, the array of ship NISC numbers would be dimensioned:

```
DIMENSION SID (SDIM)
```

Changing the parameter setting in subroutine INIDIM would automatically change the dimension of all arrays which use the SDIM parameter. It is essential that subroutine INIDIM be included in any program containing commons and that it precede the common.

The SAIL system subscripts are given in Table 5.4.

Table 5.4. Array Dimension Sizes

Dimension Variable Name	Description
CDIM	This is determined by the number of cargos, or movement records, on the CRGODAT file. The size must be sufficient to "split" cargos with intermediate stops into two records (see Paragraph 5.1.3.1).
SDIM	The maximum number of ships in the ship file (SHIPDAT).
PDIM	The number of ports in the PORTDAT file.
CHDIM	The number of channels, or aggregations of cargo. This dimension should be several times smaller than CDIM; experience will permit refinement of the size.
CTDIM	The number of cargo classes, or types of cargo, such as breakbulk, ammunition, petroleum, etc.
STDIM	The number of stowage types: breakbulk, square-feet, container, passenger, or petroleum.
TDIM	The number of days in the schedule.
INTDIM	The number of intervals into which the days in the schedule are divided.
TRDIM	The maximum number of trips made by any ship.
ROWDIM	The number of rows in the transportation problem, that is, the maximum number of individual ship compartments for any one problem type in the plan plus one for the dummy ship. Note that petroleum shipments form one problem type and currently all other cargo form a separate problem type. This dimension needs to handle only the larger problem type, not the sum of the two problems.
COLDIM	The number of columns in the transportation problem. This is the maximum number of channels in a single problem, either petroleum or dry cargo. This dimension need not exceed the number of channels (plus one for the "dummy channel").
LNKDIM	The number of records in the LINKDAT file, plus the number of links added for ports (see Paragraphs 5.1.4.1 and 5.1.4.7). The dimension will not exceed the number of links plus the number of ports.
XDIM	The length of the transportation problem solution vector. This dimension should be set to ROWDIM plus COLDIM.
FDIM	The number of types of facilities at ports, such as A=ammunition, B=container ammunition, C=container, G=general, P=petroleum. The facility subscripts are position dependent, that is, subscript 1 implies ammunition, 2 implies container ammunition, etc. FDIM should probably remain at 5.

Table 5.4. (Continued)

Dimension Variable Name	Description
SHTDIM	The number of ship types (breakbulk, RO/RO, container, passenger, etc.)
FLTDIM	The number of fleets.
PTDIM	The number of problem types in the Ship Routing Subsystem; this is "fixed" in the coding and is currently equal to 2. All dry cargo is organized into problem type 1; petroleum movements form problem type 2.
LINDIM	The number of "lines" in the LINEDAT file (see Paragraphs 4.2.3 and 5.1.4.17).
CANDIM	The number of canal status changes, currently 5. This value must be consistent with the definition of the PLANDAT file.
ATTDIM	The number of attrition rate changes, currently 6. This value must be consistent with the definition of the PLANDAT file.
CHSDIM	The maximum number of channels served by a ship on a single trip.
EVDIM	The maximum number of channel-related events (loads, intermediate stops, or unloads) on all trips of a ship.
COSDIM	The maximum number of cargos on one trip of a ship; currently set to 200. This number will need to be refined after experience with real data.
RECDIM	The maximum number of records in the LISTOUT file; one record is needed for each cargo load and unload, including intermediate stops. This dimension should be about 2.5 times CDIM.
KDIM	The maximum number of ships active at a port facility.
NODDIM	The number of nodes in the NODEDAT file, plus the number of nodes added for ports. The dimension will not exceed the number of nodes plus the number of ports.
DISDIM	The number of distances in the port-to-port distance array; this dimension needs to be at least equal to (PDIM)*(PDIM-1)/2.

5.5 PROGRAMMING SUBSCRIPT CONVENTIONS

The elements of vectors and arrays in the SAIL model, as in any FORTRAN-coded model, are reached with integer-valued subscripts. Thus, various nonnumerical data entities, such as ship types, cargo classes, and port facility types, are mapped to integers. Tables 5.5 to 5.8 show the mapping for several such quantities.

Table 5.5. Ship Type to Integer Map

Ship Type Code	Ship Type Description	Subscript Assigned
11	Breakbulk	1
15	Seatrain	2
21	RO/RO	3
43	Self-sustaining container - RO/RO	4
42	Self-sustaining container	5
14	LST	6
51	Passenger	7
61	Tanker - Shallow draft	8
62	Tanker - Small	9
63	Tanker - Handy size	10
64	Tanker - Medium	11
65	Tanker - Large	12
41	Non-self-sustaining container	13
31	LASH	14
32	Seabee	15
24	Fast Sealift	16
12	Self-sustaining container - breakbulk	17
13	Tug Barge - Dry	18
68	Tug Barge - Liquid	19
44	Non-self-sustaining container - RO/RO	20

Table 5.6. Cargo Class to Integer Map

Cargo Class Description	Character Code	Integer Assigned
Breakbulk - General	'10'	1
Breakbulk - Ammunition	'A0'	2
Breakbulk - Oversized	'1V'	3
Non-self deployable aircraft	'2N'	4
Vehicles	'30'	5
Container - General	'50'	6
Container - Vehicle	'5V'	6
Refrigerated container (REEFR)	'5R'	6
Container ammunition	'5A'	7
Floating craft	'2F'	8
Passengers (PAX)	'70'	9
Medical evacuees (MEDVAC)	'7M'	9
NEO	'7N'	10
Petroleum (POL)	'80'	11

Table 5.7. Port Facility Type to Integer Map

Port Facility Description	Character Code	Integer Assigned
Ammunition Loading Facility	'A'	1
Container Ammunition Facility	'B'	2
Other Container Facility	'C'	3
General Use Facility	'G'	4
Petroleum Facility	'P'	5

Table 5.8. Ship Stowage Type to Integer Map

Stow Type Description	Measure (and abbreviation)	Internal Code of Stowage Type	Internal Code of Measure
Breakbulk (B/B)	Measurement Tons (MT)	1	1
Deck Space Lift-on/Lift-off (LO/LO)	Square Feet (SQFT)	2	2
Container Space	Measurement Tons (Cont MT)	3	3
Passenger Space	Number of Passengers (PAX)	4	4
Bulk Petroleum Capacity	Hundreds of Barrels (CBBLS)	5	5
Roll-on/Roll-off (RO/RO)	Square Feet (SQFT)	6	2

Also, movement records are mapped to the integers as they are read; thus, information concerning the n-th record read will be reached with the subscript "n" throughout the model. Also, if cargo records are split due to intermediate stops, the subscript for the new movement generated will be added to the end of the list. Ships and ports are handled similarly;

information for the n-th ship or port read will be in the n-th element of the appropriate vector.

5.6 OTHER CODES AND CONVENTIONS

Table 5.9 relates descriptions of various fleets of ships to the codes associated with those fleets. Table 5.10 explains the codes used when cargos are excluded from the solution; each code corresponds to a reason for exclusion.

Tables 5.11 and 5.12 outline the cost functions for the ship-routing and ship-loading problems, respectively. These functions are always subject to change as development proceeds; the outlines given are those in effect at the time of publications.

Table 5.9. Fleet Codes

Description	Code
MSC Force (MSC)	10
U.S. Commercial (US)	16
Ready Reserve Fleet (RRF)	12
National Defense Reserve Fleet (NDRF)	18
Effective U.S. Controlled (EUSC)	20
North Atlantic Treaty Organization (NATO)	30
Sealift Readiness Program (SRP)	14
Korean (KFS)	40

Table 5.10. Exclusion Codes

Code	Description
1	Irretrievably late, that is, the current resources cannot possibly deliver cargo in an acceptable time-frame
2	Cargo can be in this category for any of several reasons: the optimization may have scheduled it on the dummy ship indicating insufficient timely resources.
3	Simulated schedule marked this cargo as unacceptably late
4	Cargo stranded at intermediate port
5	Cargo removed from schedule due to incomplete or inconsistent input data
6	Cargo undelivered due to insufficient bunkering capabilities (Note: these cargos *could* have been loaded and *will not* be unloaded)
7	Cargo removed from schedule because no appropriate shipping capacity is provided during the input phase
8	Too little port activity to warrant a port call
9	Inadequately loaded ships removed from schedule caused cargo to be undelivered

Table 5.11. Routing Cost Function

Condition	Cost Increment
LEVEL I - Timeliness	
Late > PAR22 * MXLT (maximum lateness)	Add BIGCST (infinity)
Late > .75 * MXLT	Add 360
5 < Late < .75 * MXLT	Add 288
0 < Late < 5	Add 216
Early > 5 and dependent leg	Add 144
MAXEAR > 5 and dependent leg	Add 72
On time	Add 0
LEVEL II - Ship utilization	
First use of ship	Add 36
LEVEL III - Compartment compatibility	
Unfavorable cargo stow type	Add 18
LEVEL IV - Port proximity to ship	
Port relatively far from ship	Add 12
Port moderately far from ship	Add 6
LEVEL V - Cargo quantity / ship capacity ratio evaluation	
B/B decks are never prime assignments	Add 4
1 < ratio < 1.25	Add 4
Ratio < 0.9 or ratio > 1.25	Add 2
LEVEL VI - Ammunition ship sheathing	
Ammunition assigned to unsheathed ship	Add 1

Table 5.12. General Network Cost Function

Condition	Cost Increment
LEVEL I - Timeliness	
Ship departs before cargo ALD	BIGCST (infinity)
Late > 1.5 * MXLT (maximum lateness)	BIGCST
Early > 1.5 * MAXEAR (maximum earliness)	BIGCST
Late > 40 days	Add 14 * N N = number of ships in plan
40 > Late > 20 days	Add 12 * N
20 > Late > 10 days	Add 10 * N
10 > Late > 5 days	Add 8 * N
Late < 5 days	Add 6 * N
Early > 10 days (except independent leg of intermediate stop)	Add 4 * N
5 < Early < 10 days (except independent leg of intermediate stop)	Add 2 * N
LEVEL II - Ship ranking (based on speed and capacity)	
For every ship	Add Rank
LEVEL III - Compartment compatibility	
Favorable cargo stow type	Subtract 1

ACRONYMS AND TERMS

AFOE--assault follow-on echelon

ALD--available to load date; the C-day cargo will be available to load at the port of embarkation

B/B--breakbulk

bunkering--ship refueling

C-day--offset day from the beginning of a plan

cargo class--there are currently eleven cargo categories separating cargos with varying handling and transportation needs

CBBLS--hundreds of barrels; petroleum, oil, and lubricant quantities are specified in hundreds of barrels

channels--aggregation units of cargos with similar shipping routes, resources, and times

closure date--the C-day concluding all scheduled deliveries

clusters--aggregation units of channels with similar location and timing needs

compartment--for SAIL purposes a ship's capacity is subdivided by stowage capacities: MT, SQ FT, container MT, PAX, and POL; each of these capacities on a ship is referred to as a compartment; currently SAIL will handle two compartments per ship

cost function (penalty function)--the penalty charged to the assignment of a particular channel to a particular ship compartment (see Fig. 5.3, Table 5.10, and Table 5.11)

cost plateau--this large fixed cost is added to all costs in the optimization version until a prime assignment has been made, then it is removed for the ship carrying the prime channel in order to attract as much cargo as possible while forming a ship trip

CONSOL--petroleum consolidation of load

deliberate planning--conceptual planning for contingency response; in the current context, it is the conceptual planning for sealift movements of men and materiel

demands--the movement requirements; when aggregated into channels, these are represented in the transportation problem as columns

dummy channel (dummy demand)--the nature of the transportation problem requires that the sum of the demands must be equal to the sum of the supplies; since the total amount of shipping needs does not exactly match the total shipping capacity, a special column is constructed representing the unused space on ships in an optimization solution

dummy ship (dummy supply)--a special row added to the transportation problem (see dummy channel) to represent uncarried cargos

execution planning--the ongoing planning process that would accompany an actual deployment; execution planning differs from deliberate planning in the requirement to make adjustments for the prevailing situation and by increased time-criticality general network problem--a special form of linear optimization well-suited to finding loads for ships for which routes have been established (see Appendix D)

FAST--Flow and Analysis System for TRANSCOM; an interactive, micro-processor based transportation system which includes SAIL as its sealift analyzer

heuristic--a rule-based system not necessarily leading to an optimum solution

LAD--latest arrival date; the C-day of the latest on-time arrival of cargo at a port of debarkation

leg--any one direct port-to-port trip segment

linear optimization--the process of maximizing (or minimizing) a linear objective function subject to a set of linear constraints

lines--connection of a port or ship location onto the ocean network grid is usually accomplished by linking to the nearest node; sometimes, however, this would lead to portages; lines are designed to prevent crossing land while making this initial connection; a future application may be in the area of ocean avoidance

links--connections between nodes which define valid paths of travel for ships; associated with each link is the great circle distance in nautical miles

LO/LO--lift on/lift off loading

MAC--Military Airlift Command

miles--in this report, miles is used interchangeably with nautical miles

MOPX--Mobility Planning and Execution; MSC's sealift planning development for both deliberate and execution planning

MSC--Military Sealift Command

MT--measurement tons, a cargo unit of volume of 100 cubic feet

MTMC--Military Transportation Management Command

NARDAC--Navy Regional Automation Center; in this report, the center in Washington is implied

NEO--Non-Combatant Evacuees

nodes--point locations on the surface of the earth specified by latitude-longitude pairs

PAX--passengers

POD--port of debarkation

POE--port of embarkation

POI--intermediate port

POL--petroleum, oil, and lubricants

prime assignment--the first channel chosen to be placed on a ship during routing; all other channels added to the ship must then be compatible with this prime

RLD--ready to load date; C-day of cargo availability at the origin

RO/RO--roll-on/roll-off loading

SAIL--Scheduling Algorithm to Improve Lift; the scheduling portion of SEASTRAT

SEACOP--Strategic Sealift Contingency Planning System; an existing sealift scheduler using heuristics

SEASTRAT--Sealift Strategic Analysis Subsystem

secondary channels--channels added to a potential ship trip acceptably compatible with the prime channel

sheathing--lining ships for safer handling of ammunition

ship trip--a single trip on a ship; it may involve more than one port call for loading and more than one port call for unloading

solution vector--an optimization solution; each solution element is represented by three variables, an amount (stored in the X vector), the column of the assignment (stored in the PC vector), and the row of the assignment (stored in the PR vector); this information describes the solution assignments of channel quantities to ship compartments

stow adjusted quantity--a representation of the space required for loading a cargo quantity

stow factors--the fraction of a particular cargo class that can be stowed in one unit of capacity of a given ship compartment

stow type--in SAIL, all ship compartments are represented as one of five types: breakbulk (in measurement tons), square feet space, container space (in measurement tons), passenger space, or bulk petroleum space (in hundreds of barrels)

supplies--ship compartment capacities represented as rows in the transportation problem

TCC--Transportation Component Command

transportation problem--a special case of linear optimization used in the ship routing problem (see Appendices B and C)

USTRANSCOM--United States Transportation Command

X vector--the vector of amounts in the solution vector

REFERENCES

FAST User's Handbook (in preparation at the Oak Ridge National Laboratory for USTRANSCOM).

George E. Forsythe, Michael A. Malcolm, and Cleve B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

Joint Staff Officer's Guide, 1986, Armed Forces Staff College, Pub 1, July 1, 1986.

Mobility System Policies, Procedures and Considerations, Joint Chiefs of Staff, JCS Pub 15, September 15, 1983.

SEASTRAT Development Plan, August 28, 1981 - SAI Comsystems.

SEASTRAT Planner Parameter File, Reference Document for SEASTRAT Phase I, April 1, 1987.

SEASTRAT Ports File, Reference Document for SEASTRAT Phase I, Draft II, April 1, 1987.

SEASTRAT Scheduling Algorithm for Improving Lift (SAIL), Conceptual Data Base Design, January 31, 1982 - SAI Comsystems.

SEASTRAT Scheduling Algorithm for Improving Lift (SAIL), Structure Specification, January 31, 1982 - SAI Comsystems.

SEASTRAT Scheduling Algorithm for Improving Lift (SAIL), Functional Description, May 18,1982 - SAI Comsystems.

SEASTRAT Scheduling Algorithm for Improving Lift (SAIL), Modeling Approach Definition, Revision I, November 18, 1982 - SAI Comsystems.

SEASTRAT Scheduling Algorithm for Improving Lift (SAIL), Modeling Methodology Formulation, Revision I, March 4, 1983 - SAI Comsystems.

Strategic Sealift Contingency Planning System (SEACOP III), Program Maintenance Manual, NARDAC, Washington D.C., Document Number TK83005L MM-01, September 1981.

Suggestions for the Improvement of the MSC Strategic Sealift Contingency Planning System (SEACOP), (undated) - MSC internal point paper.

Harvey M. Wagner, *Principles of Operations Research*, Prentice-Hall, Inc., Inglewood Cliffs, New Jersey, 1975.

APPENDICES

APPENDIX A

PROGRAM FLOW DIAGRAM

The MAIN program

```
/*      This is the control program for the SAIL system.  Key variable and array
/*      initializations are made, and subroutines that invoke the major Subsystems are
/*      called.  Data passed between subsystems are in common blocks; the commons
/*      involved in each coding block are shown in Appendix I, and the relationship
/*      between coding blocks and subsystems is given in Paragraph 5.2.1.
/*
/*      Note that all subsystems contribute to an important file, EXEPOUT, which
/*      contains exception information about the entire SAIL run.

DO

        EXECUTE SUBSYSTEM INTERFACE;

                Initialize dimension sizes (by calling INIDIM);

                Read parameters (by calling INPARA);

                Read ship and port data (by calling INSHIP and INPORT);

                Mark ship-port mismatches for use in cost functions;

                Read cargo data (by calling INCGO);

                Read loading data (by calling INSTOW);

        EXECUTE SUBSYSTEM DISTANCE;

                Invoke distance routines (by calling DISCAL);

        EXECUTE SUBSYSTEM AGGREGATION;

                Eliminate cargos that cannot be delivered due to excessive lateness (by
                calling ELIM);

                Split cargos that have intermediate stops into two pseudo-cargos (by calling
                SPLIT);

                Find the time intervals (by calling INTERV);
```

Estimate queuing at ports of debarkation (by calling PODQ);

Aggregate cargos (by calling CHANEL);

Order the channels for efficient optimization (by calling CHORDR);

Estimate queuing at ports of embarkation (by calling POEQ);

EXECUTE SUBSYSTEMS SHIP ROUTING and SHIP LOADING;

Invoke routing and loading subsystems;

DOFOR (dry cargo, petroleum);

IF (previous routing solution is to be used--that is, a file name has been specified in the variable SOLGET) THEN;

Read the existing solution file (by calling REDSOL);

ELSE

IF (heuristic routing is to be used) THEN;

EXECUTE heuristic routing (by calling NCYCLE);

ELSE

EXECUTE optimal routing (by calling ACYCLE);

ENDIF

ENDIF

IF (routing solution is to be saved--that is, a file name has been specified in the variable SOLSAV) THEN;

Save routing solution (by calling WRTSOL);

ENDIF

IF (optimal loading is specified) THEN;

EXECUTE optimal loading (by calling GNC);

ENDIF

```
        ENDDO
    EXECUTE SUBSYSTEM SIMULATION;
        Invoke simulation subsystem (by calling SIMOVR);
    EXECUTE SUBSYSTEM REPORT;
        Invoke the reporting subsystem (by calling POST);
ENDDO
```

INTERFACE SUBSYSTEM

```
/*      This subsystem accepts data from the SEASTRAT data management system.
/*
/*      Major Data Input:
/*              input files from SEASTRAT
/*                      case-dependent files: cargo data, ship data, port data, parameters
/*                      semi-permanent files: geographic data, ship loading information
/*
/*      Major Output:
/*              information describing the case in FORTRAN arrays

begin in MAIN program;

DO

        initialize dimension sizes (by calling INIDIM);

        read parameter files (by calling INPARA);

        read ship file (by calling INSHIP);

        read port file (by calling INPORT);

        find ship/port mismatches due to draft, beam, or length;

        read cargo file (by calling INCGO);

        read stow factors, load and unload times, and port facility/ship type
        combinations (by calling INSTOW);

ENDDO
```

DISTANCE COMPUTATION SUBSYSTEM

```
/*      This subsystem computes interport distances and finds initial ship availability dates.
/*
/*      Major Data Input:
/*              node, link, and line data files
/*              ports, initial ship positions
/*              initial conditions (peace/war indicator, early warning days, delay days)
/*
/*      Major Output:
/*              distances between ports for all canal conditions
/*              initial port availability dates for all ships at all ports

begin in subroutine DISCAL, called from MAIN;

DO

        IF (recomputing of distances and ship availability dates is required--that is, if the
        variables DISGET and SADGET contain no file names) THEN;

                read the files of nodes, links, and lines used to define the
                world distance network (by calling INNODE, INLINK,
                INLINE);

                DOFOR (ports not close to existing nodes);

                        add new nodes to connect all ports to the network (by
                        calling ADPORT);

                ENDDO

                DOFOR (all combinations of canal openings and closings);

                        compute interport distances, using shortest
                        path algorithm (by calling STAR, DISFIL);

                ENDDO

                        compute ship availability dates (by calling subroutine SADS);

                DOFOR (each ship);

                        find nearest node to the ship, connect ship to
                        the network (by calling NEAR, DISCHK,
                        CANFND);
```

compute time to get the ship to each port,
continue to current destination or divert to
POEs as determined by war/peace indicator;

add delay days and subtract early warning days,
giving the initial ship availability date at each port;

ELSE

Read interport distances and ship availability
dates from the files specified in DISGET and
SADGET (by calling GETDIS);

ENDDO

ENDIF

IF (the distances and ship availability dates are to be saved--that is, if file names have been specified in variables DISSAV and SADSAV) THEN;

Write the interport distances and ship availability dates to
the specified files (by calling SAVDIS);

ENDIF

ENDDO

AGGREGATION SUBSYSTEM

```
/*      This subsystem aggregates the cargos into channels and clusters for use in the ship
/*      routing and loading subsystems.
/*
/*      Major Data Input:
/*              cargo characteristic data
/*                      cargo classes
/*                      ports of embarkation and debarkation
/*                      special handling (preferred ships,
/*                      intermediate
/*                      stops,
/*                        AFOE)
/*                      unit identification
/*                      availability and delivery dates
/*              parameters
/*                      time interval widths
/*
/*      Major Output:
/*              aggregations of cargo into channels
/*              aggregations of channels into clusters

begin in MAIN program

DO
        DOFOR (all cargos);

                eliminate cargos that clearly cannot be delivered within their
                maximum lateness criteria--that is, those for which there are
                no ships that can deliver them on time (by calling ELIM);

        ENDDO

        DOFOR (all cargos with intermediate stops);

                "split" cargos into two trip "legs" (by calling SPLIT);

        ENDDO

        initialize queuing arrays;

        find the time intervals used in aggregation (by calling INTERV);

        estimate queuing at ports of debarkation (by calling PODQ);

        DO
```

```
        channel aggregations (by calling subroutine CHANEL);

        initialize aggregation (channel) arrays (by calling INITCH);

        DO (for each cargo);

            estimate required load time;

            IF (there is already a channel that has this
            cargo's port of embarkation, port of
            debarkation, cargo class and unit or non-unit,
            and has a similar shipping date and available
            to load date) THEN;

                add this cargo to the channel;

            ELSE

                start a new channel with this cargo;

            ENDIF

            find clusters of channels with similar
            characteristics;

            return to MAIN;

        ENDDO

    ENDDO

    estimate queuing times at ports of embarkation (by calling POEQ);

ENDDO
```

SHIP ROUTING SUBSYSTEM
(Optimization-based Process)

```
/*    This subsystem uses a process that interleaves linear optimization and decision rules
/*    to find routes for ships.
/*
/*    1.        A transportation problem is constructed using the channels as sources and
/*              the ship compartments as sinks; costs represent the effectiveness of a ship
/*              carrying a channel
/*
/*    2.        The transportation problem is solved
/*
/*    3.        From the solution, a single good assignment of a channel to a ship is
/*              chosen to form the seed of a ship trip; this is called the prime channel
/*
/*    4.        Other channels compatible with the prime are added to the ship until there
/*              are no other viable candidates or until the ship is full
/*
/*    5.        The optimization problem is revised to reflect the assignments made and
/*              the ship's route
/*
/*    6.        The transportation problem is resolved, beginning with the previous
/*              solution
/*
/*    7.        If not finished, return to 3, above
/*
/*      Major Input Data:
/*              ship descriptions and availability
/*              channel attributes
/*                      ports involved
/*                      time requirements
/*                      cargo characteristics
/*
/*      Major Output:
/*              a loading of channels to ships that implies routes for the ships

begin in subroutine ACYCLE, called from RTLOAD, called from MAIN;

DOFOR (dry cargo and petroleum);

        set up the transportation problem;

                DOFOR (all ship-trip compartments which carry cargo in this problem);

                        set up ship capacities using approximate stow factors;

                        find cost of each cargo aggregation on each ship-trip compartment
```

```
            (by calling MATGEN);

        ENDDO
    ENDDO

    find an initial feasible solution to the transportation problem (by calling FIRSTX);

    find an initial optimal solution (by calling SOLVE);

    set the current trip number for each ship to one;

    DOWHILE (channels remain to be placed on routed ships);

        in subroutine APRIME, called from ACYCLE;

        DOFOR (the current trip of each ship);

        find the best aggregation of cargo assigned to the ship, considering both
        cargo size and timing of shipment;

        find the largest of these best assignments;
        this load will be the nucleus, or prime, load for generating a route;

        determine timeliness of the delivery of the prime cargo (using CHINIT);

        IF (the prime cargo will be delivered late) THEN;

            give the assignment a high price;

            re-optimize;

        ELSE

            determine geographic compatibility of other assignments on the ship
            trip with the prime assignment (by calling SECON1);

            DOFOR (incompatible assignments);

                assign a high price (by calling MATREV);

                re-optimize (by calling SOLVE);

            ENDDO

            DOFOR (the remaining candidate assignments);

                find the one that causes the least increase in routing distance;
```

determine timing compatibility of the candidate with the prime assignment;

mark the unsuccessful candidates so they will receive a high cost in the next re-optimization (by calling CHEVAL);

place the successful candidates on ship using actual stow factors to find the amount to be carried (by calling ADDCHN);

ENDDO

return to subroutine ACYCLE;

determine if the sailed ship is attrited (by calling function RISK);

increment the current trip number of the sailed ship;

compute new ship availability dates for the next trip of the sailed ship (by calling ASAILM);

delete routed cargos from the transportation problem (by calling PURGE);

compute new costs for the sailed ship which reflect the ship availability dates for the next trip (by calling MATREV);

re-optimize (by calling SOLVE);

ENDDO

see if all cargo is on sailed ships, or cannot be scheduled (by calling DONE);

ENDDO

ENDDO

SHIP ROUTING SUBSYSTEM
(Heuristic-based Process)

```
/*    This subsystem routes ships using decision rules.
/*
/*    1.      The cost matrix of the transportation problem is formed as in the
/*            optimization-based method
/*
/*    2.      The next most critical cargo to move is chosen based on quantity and
/*            urgency
/*
/*    3.      Channels clustered with that critical channel are marked
/*
/*    4.      The best ship to move that cargo is chosen, based primarily on the cost
/*            matrix
/*
/*    5.      Other ships nearly as good as the best are found
/*
/*    6.      The marked channels are trial loaded on each of the candidate ships; a
/*            score is computed based on how many of the channels are carried and how
/*            much space is left on the ship
/*
/*    7.      The ship with the best score is loaded and sailed
/*
/*    8.      The process continues until all channels are routed or cannot be moved
/*
/*      Major Input Data:
/*            ship descriptions and availability
/*            channel attributes
/*                  ports involved
/*                  time requirements
/*                  cargo characteristics
/*
/*      Major Output:
/*            a loading of channels to ships that implies routes for the ships

begin in subroutine NCYCLE, called from RTLOAD, called from MAIN;

DOFOR (dry cargo and petroleum);

/*      note that, while the transportation problem is not solved, the structure and cost
        matrices are used

        set up the transportation problem;

        find all ship-trip compartments which carry cargo in this problem;
```

find cost of each cargo aggregation on each ship-trip compartment (by calling MATGEN);

set the current trip number for each ship to one;

DOWHILE (channels remain to be loaded);

in subroutine NPRIME, called from NCYCLE

find the next most urgent (or prime) channel;

using the cost matrix, find the best ship for carrying that channel;

determine if the cargo can be delivered in time (by calling CHINIT);

IF (channel not on time) THEN;

give the assignment a high price;

ELSE

find all channels clustered with the prime;

find all ships almost as good as the best ship;

trial load the clusters of channels onto each of these ships (by calling TEMPLD);

compute a score for each ship based on the ship's ability to carry the channels and the amount of space left over on the ship;

select the best of the ships;

load the prime channel on that ship (by calling STASH);

check to see if the ship is full (by calling function FULL);

DOWHILE (ship is not full and compatible channels remain);

determine geographic compatibility of other assignments on the ship trip with the prime assignment (by calling SECON1);

for incompatible assignments, assign a high price (by calling MATREV);

find the next best channel (by calling NXTBST);

```
            ENDDO
        ENDIF
        return to NCYCLE;
        determine if the sailed ship is attrited (by calling function RISK);
        increment the current trip number of the sailed ship;
        compute new ship availability dates for subsequent trips of the sailed ship
        (by calling NSAILM);
        compute new costs for the sailed ship which reflect the ship availability
        dates for the next trip (by calling MATREV);
    ENDDO
        build a solution vector (by calling BLDSOL);
ENDDO
```

SHIP LOADING SUBSYSTEM

```
/*      This subsystem uses linear programming to allocate channels to the ships with
/*      routes established by the ship routing subsystem using actual stow factors.
/*
/*      Major Data Input:
/*              routes from the ship routing subsystem
/*              ship data
/*              channel information
/*
/*      Major Outputs:
/*              stow-adjusted assignments of channels to routed ships to be used by the
/*              simulation
/*              subsystem
```

DO

simulate trips of the routed ships (by calling SIMULA...described in the Simulation Subsystem below);

find arrival and departure times of all ships at ports on route (by calling ARRDEP);

use these times to compute new costs for assigning cargo to the routed ships (by calling function GTCOST);

set up a generalized network problem using these costs and precise stow factors;

solve the generalized network problem (use routines of GENNET);

use the solution of the network problem as input to the simulation system;

ENDDO

SIMULATION SUBSYSTEM

```
/*      This subsystem simulates the loading, sailing, and unloading of ships using the
/*      assignments from the ship loading subsystem. This is done at the more detailed
/*      level of cargo data, as opposed to the aggregated channel data used in the routing
/*      and loading. Special routines can be invoked to compute bunkering requirements.
/*
/*      Major Input Data:
/*              channel to ship assignments
/*              channel information
/*              cargo data
/*              ship data
/*              port data
/*              loading and unloading data
/*              distance information
/*
/*      Major Outputs:
/*              detailed ship itinerary and cargo exclusions, the LISTOUT and EXCLOUT
/*              files, which are converted to the final FINIOUT file by the reporting
/*              subsystem bunker data, if requested
```

in subroutine SIMOVR, called from MAIN

DOWHILE (any of the following conditions prevail);

cond1: this is the first pass
cond2: late cargo was removed on the last pass
cond3: a final pass is needed to remove sparsely loaded ships

DO (by calling subroutine SIMULA);

structure the solution from the linear program as events which are to occur (by calling SKEDIN);

find the sequence of port stops to be made by each ship (by calling PATH);

find the time of the first event on each ship (which is docking at its first port of embarkation);

DOWHILE (events remain unsimulated);

find the next most imminent event;

DO CASE (cargo loading, cargo unloading, docking, sailing);

cond1 (load cargo, by calling LOADUP):

```
        determine the time to load;
        place the cargo on the ship;
        write a record to LISTOUT file;
cond2 (unload cargo, by calling DEBARK):
        determine the time to unload;
        remove cargo from the ship;
        see if the cargo meets delivery time requirements;
        DO CASE (first pass, later passes);
        cond1 (first pass):
                mark late cargo;
        cond2 (later passes):
                delete late cargo;
        ENDDO
        IF (cargo was not deleted) THEN;
                write a record to LISTOUT file;
        ELSE
                write a record to EXCLOUT file;
        ENDIF
cond3 (docking):
        IF (there is unused port capacity) THEN;
                add docking time and determine next event for the ship;
        ELSE
                delay ship docking;
        ENDIF
```

```
            cond4 (sailing):

                determine sailing time;

                determine next event for the ship;

            ENDDO
        ENDDO
    ENDDO
```

REPORTING SUBSYSTEM

```
/*      This subsystem reads the exclusion and ship itinerary files (EXCLOUT and
/*      LISTOUT), removes slack from the schedule, and writes the final ship itinerary file
/*      (FINIOUT). It also computes summary information about the schedule and prints
/*      reports.
/*
/*      Major Input Data:
/*              itinerary file from the Simulation Subsystem (LISTOUT)
/*              exclusion file from the Simulation Subsystem (EXCLOUT)
/*              cargo, ship, and port data
/*              bunkering information
/*
/*      Major Outputs:
/*              summary reports
/*              final itinerary file (FINIOUT)

in subroutine POST, called from MAIN;

DO

        read exclusion and itinerary files (EXCLOUT and LISTOUT);

        remove slack from the schedule;

        write final itinerary file (FINIOUT);

        compute summary information;

        write summary reports:

                histograms;
                cargo report (by calling REPTC);
                ship report (by calling REPTS);
                port report (by calling REPTP);
                bunker report;

ENDDO
```

APPENDIX B

LINEAR PROGRAMMING: A NONTECHNICAL INTRODUCTION

Linear programming is one of several operations research techniques called "resource allocation" methods. The basic idea is to assign a set of scarce resources to several competing in uses a way that best accomplishes some objective. Generally, there are limitations on the way the resources can be assigned. These limitations are called "constraints."

These problems are best understood by looking at an example. Consider the case of the famous violin maker, Signor A. Stradivari. His resources were fine woods and his time, both in limited supply. He produced two types of instruments: violins and cellos. Violins were the more tedious to make and therefore took more time; cellos obviously required more wood. The Signor could make a profit on both types of instruments. His problem was to figure out the best mix of instruments to maximize his profits. Unfortunately, linear programming, which is ideally suited to this problem, would not be invented for 250 years... Signor Stradivari thus declared bankruptcy and went into business scheduling military shipments for the Genoans; unfortunately, his technique for making fine instruments was lost forever.

But had linear programming been around, the problem could have been formulated as follows. Suppose it took 4.5 weeks to produce a violin and 3 weeks to produce a cello. After festivals and his annual visit to his in-laws, Signor Stradivari had 45 work weeks in each year. The Musical Signor's Cooperative (MSC) allocated 10 kilos of fine wood to him each year; it took half a kilo to make a violin and 2 kilos to make a cello. Violins and cellos gave him a profit of 75 and 100 lira, respectively.

Now for the mathematics: let V and C stand for the number of violins and cellos to produce each year. The objective is to maximize the profit, that is

$$\text{maximize} \quad 75*V + 100*C.$$

But the constraints on time and wood must not be violated. The time constraint, in mathematical terms, is

$$4.5*V + 3*C \leq 45,$$

which can be read as "4.5 times the number of violins plus 3 times the number of cellos must be less than or equal to 45 weeks." Similarly, the constraint on wood can be written

$$.5*V + 2*C < 10.$$

Linear programming also imposes two "hidden" constraints: neither V nor C may be negative.

The problem is now completely formulated. It is solved using a computer technique developed about 35 years ago which starts with a trivial solution ($V = 0$ and $C = 0$) and keeps finding new solutions until the best solution is found. The method is surprisingly efficient; the key to this is that it always goes to a better solution at each step. The answer that would be found for our musical problem is to produce 8 violins and 3 cellos.

The constraints can be checked to see that this solution honors them. Eight violins would take 36 weeks to produce; 3 cellos would take another 9 weeks. The total time is the 45 weeks available. Similarly, the solution uses all of the allotted wood. (One should note that, at some relative profits, the best solution might call for using less than all the wood or less than all the time available.) It is easily seen that Signor Stradivari's profit in the case described would be 900 lira.

The sealift scheduling problem is similar, only in this case we are minimizing costs instead of maximizing profits. The resources to be allocated are ships; the decisions are to find which cargos should be placed on each ship. Variables are set up only for cargos picked up and delivered along the route the ship will take. There is some cost (or penalty) associated with placing a particular cargo on any given ship; this cost is related mainly to how timely the ship can deliver the cargo. The objective is to find the combinations of cargos and ships that minimize the total shipping cost. Constraints are written: don't ship more cargo than you have, and don't put more cargo on a ship than it will hold.

The "transportation problem" used in the routing portion of the SAIL model is a special case of linear programming. The advantage of using this formulation is that large problems can be solved much faster than if the more general linear programming formulation were used. To be a transportation problem, our routing problem must have the sum of all cargos to be shipped equal to the sum of shipping available, and all the coefficients of the constraints must be 1. The first condition is easily met by adding dummy ships and dummy cargos with capacities chosen to make the supply and demand balance. The second condition is met by using approximations to the stow factors.

APPENDIX C

SHIP ROUTING: A TRANSPORTATION PROBLEM FORMULATION

To form the demands (or columns):

Let A_j be the amount of channel j to be shipped.

Form D_j as A_j* (cargo-related stow factor for the cargo class of channel j). Keep the pointer

ch_j = channel number of column j.

To form the supplies (or rows):

Let $K_{s,st}$ be the capacity of ship s, compartment st.

Form S_i, the supply in row i as

$K_{s,st}$* (capacity-related stow factor for compartment st of the ship type of ship s).

Keep the pointers

sh_i = ship number of row i,
tr_i = trip number of row i, and
st_i = stowage compartment type of row i.

Add a dummy column (representing unused ship space):

D_n = large amount (this amount should never be an active constraint).

Add a dummy row (representing channel amounts not scheduled on the next trip of all ships):

$$S_m = \sum_{j=1}^{n} D_j - \sum_{i=1}^{m-1} S_i \quad .$$

As discussed in Paragraph 4.2.5.1, the problem is modified each time a ship sails; the channel amounts placed on the ship are decremented, and the ship's trip number is incremented. Taking the sailed quantities out of the problem must be done carefully to maintain the structure of the transportation problem; the strategy used is outlined in Paragraph 5.1.5.12.

APPENDIX D

SHIP LOADING: A LINEAR PROGRAMMING FORMULATION

Let x_{ij} be the amount of demand j carried on supply (resource) i, where the demands and supplies are as defined in the transportation problem (Appendix A). A variable is formed for each feasible channel-ship trip combination. Note that fixing the routes greatly restricts the number of variables formed.

Let c_{ij} be the cost of the assignment of demand j to supply i,

s_{ij} be the stow factor for a unit of demand j on supply i,

K_i be the capacity of supply i, and

A_j be the amount of demand j.

Objective: to maximize feasible deliveries at lowest cost

$$\text{Min} \sum_i \sum_j c_{ij} x_{ij}$$

subject to: limitations on shipping capacities

$$\sum s_{ij} x_{ij} \leq K_i, \text{ for all } i,$$

and limitations on amounts of cargo

$$\sum x_{ij} = A_j, \text{ for all } j.$$

Feasibility is ensured by including a large phantom ship with high assignment costs.

This problem formulation fits a special case of linear programming called the generalized network problem; a solution process more efficient than that for general linear programming can therefore be used.

APPENDIX E

TABLES OF PORT FACILITIES, LOAD RATES, UNLOAD RATES, STOW FACTORS, AND CARGO/SHIP TYPE ASSIGNMENTS

The tables in this appendix show relationships between the various classes of cargo and the different compartments of the several ship types. The first table shows the types of port facilities required to load or unload a cargo class on each ship compartment. Where two types are shown, the first is preferred. The next two tables show the loading and unloading rates of cargo classes and ship compartments.

The stow factor table is more complicated. The numbers represent the efficiency of stowing each cargo class in each compartment type. For example, only 45 measurement tons of vehicles could be placed in 100 measurement tons of breakbulk hold space. Some cargos pack with more than 100% efficiency, though: 110 square feet of aircraft can typically be fitted into 100 square feet of RO/RO space. Another efficiency involves containers, which, when loaded into compartments measured in square feet, can be stacked two deep. These cases are indicated by asterisks; the effective stow factor is therefore twice the number in the table.

The stow factor table has an extra column and an extra row. These are overall stow factors for the ship compartments and cargo classes, respectively, and are used in the transportation problem of the ship routing subsystem. The product of a ship and cargo factor should approximate the true stow factor given in the table; obviously, some of the products come closer than others.

The final table in this appendix gives the desirability of placing each cargo class into each type of ship compartment; an entry of 1 is most favorable, 2 less favorable, and 3 least favorable. No entry means the combination is infeasible. The use of this information had not been incorporated into the SAIL model at the time of publication of this report.

The ship type definitions contained in these tables were derived from the SEASTRAT/JSCP MODULE - SHIP FILE FIELD STRUCTURE dated 7 Feb 1990. The question marks in the tables indicate combinations of ships and cargos for which information has not been received.

JSCP PORT FACILITIES REQUIRED FOR CARGO-CLASS/SHIP-TYPE COMBINATIONS

SHIP TYPE	SHIP COMPARTMENT STOW UNITS	CARGO CLASS											
		1 10 B/B GEN	2 A0 B/B AMMO	3 1V B/B OVSZ	4 2N NSDA	5 30 VEHICLES	6 C GEN 50 C VEH 5V REEFR 5R	7 5A CT AMMO	8 2F FLT CRFT	9 70 PAX	10 7M MEDVAC	11 7N NEO	12 80 POL
1 11 BREAKBULK	B/B M/T DECK LO/LO	G G	A N/A	N/A G	G G	G G	N/A C,G	N/A N/A	N/A G	N/A N/A	N/A N/A	G N/A	N/A N/A
2 12 SS CNTR-B/B	CNTR M/T B/B M/T	N/A G	N/A A	N/A N/A	N/A G	N/A G	C,G N/A	B,A N/A	N/A N/A	N/A N/A	N/A N/A	N/A G	N/A N/A
3 TUG BARGE 13 DRY	B/B M/T	G	A	N/A	N/A	N/A	C	N/A	N/A	N/A	N/A	G	N/A
4 14 LST	B/B M/T LO/LO	G N/A	A N/A	N/A G	N/A G	N/A G	N/A C	N/A N/A	N/A N/A	N/A N/A	N/A N/A	G N/A	N/A N/A
5 15 SEATRAIN	LO/LO	G	A	G	G	G	C	B	G	N/A	N/A	G	N/A
6 21 RO/RO	RO/RO	N/A	N/A	G	G	G	C	B	N/A	N/A	N/A	G	N/A
7 24 FAST SEALIFT	B/B M/T CNTR M/T RO/RO LO/LO	? G N/A ?	? A N/A ?	? N/A G ?	? N/A G ?	? G G ?	? N/A C ?	? N/A N/A ?	? N/A N/A ?	? N/A N/A ?	? N/A N/A ?	? G * N/A * ?	? N/A N/A ?
8 25 MARITIME PREP	RO/RO LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
9 31 LASH	B/B M/T CNTR M/T	G ?	A ?	G ?	G ?	G ?	C ?	B ?	G ?	N/A ?	N/A ?	G ?	N/A ?
10 32 SEABEE	B/B M/T CNTR M/T	G ?	A ?	G ?	G ?	G ?	C ?	B ?	G ?	N/A ?	N/A ?	G ?	N/A ?
11 41 NSS CONTAINER	CNTR M/T	N/A	N/A	N/A	N/A	N/A	C	B	N/A	N/A	N/A	G *	N/A
12 42 SS CONTAINER	CNTR M/T	N/A	N/A	N/A	N/A	N/A	C,G	B,A	N/A	N/A	N/A	G	N/A

Facility types: A=ammo, B=container ammo, C=container, G=general, P=POL
* Entries consistent with load/unload tables, but at odds with 5/19/87 K. Wiswesser letter

JSCP PORT FACILITIES REQUIRED FOR CARGO-CLASS/SHIP-TYPE COMBINATIONS (CONTINUED)

SHIP TYPE	SHIP COMPARTMENT STOW UNITS	CARGO CLASS 1 10 B/B GEN	2 A0 B/B AMMO	3 1V B/B OVSZ	4 2N NSDA	5 30 VEHICLES	6 C GEN 50 C VEH 5V REEFR 5R	7 5A CT AMMO	8 2F FLT CRFT	9 70 PAX	10 7M MEDVAC	11 7N NEO	12 80 POL
13 43 SS CNTR-RO/RO	CNTR M/T RO/RO	N/A N/A	N/A N/A	N/A G	N/A G	N/A G	C,G C	B,A N/A	N/A N/A	N/A N/A	N/A N/A	G * N/A	N/A N/A
14 46 TACS-AUX CRANE	CNTR M/T	?	?	?	?	?	?	?	?	?	?	?	?
15 48 NSS MOD CNTR	CNTR M/T LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
16 49 SS MOD CNTR	CNTR M/T LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
17 51 PASSENGER	PAX	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	G	G	G	N/A
18 TANKER 61 SHALLOW DRAFT	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	P
19 TANKER 62 SMALL	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	P
20 TANKER 63 HANDY SIZE	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	P
21 TANKER 64 MEDIUM	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	P
22 TANKER 65 LARGE	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	P
23 TUG BARGE 68 LIQUID	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	P

Facility types: A=ammo, B=container ammo, C=container, G=general, P=POL
* Entries consistent with load/unload tables, but at odds with 5/19/87 K. Wiswesser letter

JSCP UNLOADING RATES PER DAY

SHIP TYPE	SHIP COMPARTMENT STOW UNITS	CARGO CLASS 1 10 B/B GEN	2 A0 B/B AMMO	3 1V B/B OVSZ	4 2N NSDA	5 30 VEHICLES	6 C GEN 50 C VEH 5V REEFR 5R	7 5A CT AMMO	8 2F FLT CRFT	9 70 PAX	10 7M MEDVAC	11 7N NEO	12 80 POL
1 11 BREAKBULK	B/B M/T DECK LO/LO	3100 MT 3100 MT	2000 MT N/A	N/A 3100 MT	3100 MT 3100 MT	3100 MT 3100 MT	N/A 9700 MT	N/A N/A	N/A 6000 MT	N/A N/A	N/A N/A	6000PAX N/A	N/A N/A
2 12 SS CNTR-B/B	CNTR M/T B/B M/T	N/A 3100 MT	N/A 2000 MT	N/A N/A	N/A 3100 MT	N/A 3100 MT	16200 MT N/A	7250 MT N/A	N/A N/A	N/A N/A	N/A N/A	N/A 6000PAX	N/A N/A
3 TUG BARGE 13 DRY	B/B M/T	900 MT	600 MT	N/A	N/A	N/A	7400 MT	N/A	N/A	N/A	N/A	6000PAX	N/A
4 14 LST	B/B M/T LO/LO	3100 MT N/A	2000 MT N/A	N/A 3100 MT	N/A 3100 MT	N/A 3100 MT	N/A 9700 MT	N/A N/A	N/A N/A	N/A N/A	N/A N/A	6000PAX N/A	N/A N/A
5 15 SEATRAIN	LO/LO	6900 MT	4500 MT	6900 MT	6900 MT	6900 MT	9700 MT	7250 MT	11000 MT	N/A	N/A	6000PAX	N/A
6 21 RO/RO	RO/RO	N/A	N/A	26600 MT	26600 MT	33500 MT	33500 MT	9660 MT	N/A	N/A	N/A	6000PAX	N/A
7 24 FAST SEALIFT	B/B M/T CNTR M/T RO/RO LO/LO	? 3100 MT N/A ?	? 2000 MT N/A ?	? N/A 26600 MT ?	? N/A 16625 MT ?	? 15600 MT 20750 MT ?	? N/A 9700 MT ?	? N/A N/A ?	? N/A N/A ?	? N/A N/A ?	? N/A N/A ?	? 6000PAX N/A ?	? N/A N/A ?
8 25 MARITIME PREP	RO/RO LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
9 31 LASH	B/B M/T CNTR M/T	8100 MT ?	2000 MT ?	8100 MT ?	8100 MT ?	8100 MT ?	8100 MT ?	7250 MT ?	11000 MT ?	N/A ?	N/A ?	6000PAX ?	N/A ?
10 32 SEABEE	B/B M/T CNTR M/T	7400 MT ?	2000 MT ?	7400 MT ?	7400 MT ?	7400 MT ?	7400 MT ?	7250 MT ?	23600 MT ?	N/A ?	N/A ?	6000PAX ?	N/A ?
11 41 NSS CONTAINER	CNTR M/T	N/A	N/A	N/A	N/A	N/A	48600 MT	7250 MT	N/A	N/A	N/A	6000PAX	N/A
12 42 SS CONTAINER	CNTR M/T	N/A	N/A	N/A	N/A	N/A	48600 MT	7250 MT	N/A	N/A	N/A	6000PAX	N/A

JSCP UNLOADING RATES PER DAY (CONTINUED)

SHIP TYPE	SHIP COMPARTMENT STOW UNITS	CARGO CLASS 1 10 B/B GEN	2 A0 B/B AMMO	3 1V B/B OVSZ	4 2N NSDA	5 30 VEHICLES	6 C GEN 50 C VEH 5V REEFR 5R	7 5A CT AMMO	8 2F FLT CRFT	9 70 PAX	10 7M MEDVAC	11 7N NEO	12 80 POL
13 43 SS CNTR-RO/RO	CNTR M/T RO/RO	N/A N/A	N/A N/A	N/A 20000 MT	N/A 20000 MT	N/A 33500 MT	48600 MT 25200 MT	7250 MT N/A	N/A N/A	N/A N/A	N/A N/A	6000PAX N/A	N/A N/A
14 46 TACS-AUX CRANE	CNTR M/T	?	?	?	?	?	?	?	?	?	?	?	?
15 48 NSS MOD CNTR	CNTR M/T LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
16 49 SS MOD CNTR	CNTR M/T LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
17 51 PASSENGER	PAX	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	6000PAX	1500PAX	6000PAX	N/A
18 TANKER 61 SHALLOW DRAFT	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	240CBBL
19 TANKER 62 SMALL	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	720CBBL
20 TANKER 63 HANDY SIZE	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	960CBBL
21 TANKER 64 MEDIUM	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	960CBBL
22 TANKER 65 LARGE	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1200CBBL
23 TUG BARGE 68 LIQUID	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	960CBBL

JSCP LOADING RATES PER DAY

SHIP TYPE	SHIP COMPARTMENT STOW UNITS	CARGO CLASS 1 10 B/B GEN	2 A0 B/B AMMO	3 1V B/B OVSZ	4 2N NSDA	5 30 VEHICLES	6 C GEN 50 C VEH 5V REEFR 5R	7 5A CT AMMO	8 2F FLT CRFT	9 70 PAX	10 7M MEDVAC	11 7N NEO	12 80 POL
1 11 BREAKBULK	B/B M/T DECK LO/LO	3100 MT 3100 MT	2000 MT N/A	N/A 3100 MT	3100 MT 3100 MT	3100 MT 3100 MT	N/A 9700 MT	N/A N/A	N/A 6000 MT	N/A N/A	N/A N/A	6000PAX N/A	N/A N/A
2 12 SS CNTR-B/B	CNTR M/T B/B M/T	N/A 3100 MT	N/A 2000 MT	N/A N/A	N/A 3100 MT	N/A 3100 MT	16200 MT N/A	7250 MT N/A	N/A N/A	N/A N/A	N/A N/A	N/A 6000PAX	N/A N/A
3 TUG BARGE DRY 13	B/B M/T	900 MT	600 MT	N/A	N/A	N/A	7400 MT	N/A	N/A	N/A	N/A	6000PAX	N/A
4 14 LST	B/B M/T LO/LO	3100 MT N/A	2000 MT N/A	N/A 3100 MT	N/A 3100 MT	N/A 3100 MT	N/A 9700 MT	N/A N/A	N/A N/A	N/A N/A	N/A N/A	6000PAX N/A	N/A N/A
5 15 SEATRAIN	LO/LO	5700 MT	3700 MT	5700 MT	5700 MT	5700 MT	9700 MT	7250 MT	11000 MT	N/A	N/A	6000PAX	N/A
6 21 RO/RO	RO/RO	N/A	N/A	20000 MT	20000 MT	25200 MT	25200 MT	7250 MT	N/A	N/A	N/A	6000PAX	N/A
7 24 FAST SEALIFT	B/B M/T CNTR M/T RO/RO LO/LO	? 3100 MT N/A ?	? 2000 MT N/A ?	? N/A 12500 MT ?	? N/A 12500 MT ?	? 15600 MT 15600 MT ?	? N/A 9700 MT ?	? N/A N/A ?	? N/A N/A ?	? N/A N/A ?	? N/A N/A ?	? 6000PAX N/A ?	? N/A N/A ?
8 25 MARITIME PREP	RO/RO LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
9 31 LASH	B/B M/T CNTR M/T	8100 MT ?	2000 MT ?	8100 MT ?	8100 MT ?	8100 MT ?	8100 MT ?	7250 MT ?	11000 MT ?	N/A ?	N/A ?	6000PAX ?	N/A ?
10 32 SEABEE	B/B M/T CNTR M/T	7400 MT ?	2000 MT ?	7400 MT ?	7400 MT ?	7400 MT ?	7400 MT ?	7250 MT ?	23600 MT ?	N/A ?	N/A ?	6000PAX ?	N/A ?
11 41 NSS CONTAINER	CNTR M/T	N/A	N/A	N/A	N/A	N/A	48600 MT	7250 MT	N/A	N/A	N/A	6000PAX	N/A
12 42 SS CONTAINER	CNTR M/T	N/A	N/A	N/A	N/A	N/A	48600 MT	7250 MT	N/A	N/A	N/A	6000PAX	N/A

JSCP LOADING RATES PER DAY (CONTINUED)

SHIP TYPE	SHIP COMPARTMENT STOW UNITS	CARGO CLASS											
		1 10 B/B GEN	2 A0 B/B AMMO	3 1V B/B OVSZ	4 2N NSDA	5 30 VEHICLES	6 C GEN 50 C VEH 5V REEFR 5R	7 5A CT AMMO	8 2F FLT CRFT	9 70 PAX	10 7M MEDVAC	11 7N NEO	12 80 POL
13 43 SS CNTR-RO/RO	CNTR M/T RO/RO	N/A N/A	N/A N/A	N/A 20000 MT	N/A 20000 MT	N/A 25200 MT	48600 MT 25200 MT	7250 MT N/A	N/A N/A	N/A N/A	N/A N/A	6000PAX N/A	N/A N/A
14 46 TACS-AUX CRANE	CNTR M/T	?	?	?	?	?	?	?	?	?	?	?	?
15 48 NSS MOD CNTR	CNTR M/T LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
16 49 SS MOD CNTR	CNTR M/T LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
17 51 PASSENGER	PAX	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	6000PAX	1500PAX	6000PAX	N/A
18 TANKER 61 SHALLOW DRAFT	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	240CBBL
19 TANKER 62 SMALL	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	720CBBL
20 TANKER 63 HANDY SIZE	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	960CBBL
21 TANKER 64 MEDIUM	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	960CBBL
22 TANKER 65 LARGE	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1200CBBL
23 TUG BARGE 68 LIQUID	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	960CBBL

JSCP STOW FACTORS AS A FRACTION OF USABLE CARGO SPACE

SHIP TYPE	SHIP COMPARTMENT STOW UNITS	CARGO CLASS 1 10 B/B GEN	2 A0 B/B AMMO	3 1V B/B OVSZ	4 2N NSDA	5 30 VEHICLES	6 C GEN 50 C VEH 5V REEFR 5R	7 5A CT AMMO	8 2F FLT CRFT	9 70 PAX	10 7M MEDVAC	11 7N NEO	12 80 POL	SHIP STOW FACTOR
1 11 BREAKBULK	B/B M/T DECK LO/LO	.80 .80	.65 N/A	N/A .75	.45 .75	.45 .75	N/A .80 *	N/A N/A	N/A .80	N/A N/A	N/A N/A	.01 N/A	N/A N/A	.65 .80
2 12 SS CNTR-B/B	CNTR M/T B/B M/T	N/A .80	N/A .65	N/A N/A	N/A .45	N/A .45	.80 N/A	.50 N/A	N/A N/A	N/A N/A	N/A N/A	N/A .01	N/A N/A	.80 .80
3 TUG BARGE 13 DRY	B/B M/T	.80	.65	N/A	N/A	N/A	.80	N/A	N/A	N/A	N/A	.01	N/A	.70
4 14 LST	B/B M/T LO/LO	.90 N/A	.65 N/A	N/A .90	N/A .90	N/A .90	N/A .90 *	N/A N/A	N/A N/A	N/A N/A	N/A N/A	.01 N/A	N/A N/A	.85 .90
5 15 SEATRAIN	LO/LO	.75	.65	1.1	1.1	.75	.75 *	.50 *	.75	N/A	N/A	.01	N/A	.80
6 21 RO/RO	RO/RO	N/A	N/A	1.1	1.1	.75	.75 *	.50 *	N/A	N/A	N/A	.01	N/A	1.15
7 24 FAST SEALIFT	B/B M/T CNTR M/T RO/RO LO/LO	? .75 N/A ?	? .65 N/A ?	? N/A 1.1 ?	? N/A 1.1 ?	? .45 .75 ?	? N/A .90 * ?	? N/A N/A ?	? N/A N/A ?	? N/A N/A ?	? N/A N/A ?	? .01 N/A ?	? N/A N/A ?	.70 .70 .80 .80
8 25 MARITIME PREP	RO/RO LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
9 31 LASH	B/B M/T CNTR M/T	.80 ?	.65 ?	.80 ?	.80 ?	.55 ?	.80 ?	.50 ?	.80 ?	N/A ?	N/A ?	.01 ?	N/A ?	.75 .70
10 32 SEABEE	B/B M/T CNTR M/T	.80 ?	.65 ?	.80 ?	.80 ?	.55 ?	.80 ?	.50 ?	.80 ?	N/A ?	N/A ?	.01 ?	N/A	.75 .70
11 41 NSS CONTAINER	CNTR M/T	N/A	N/A	N/A	N/A	N/A	.80	.50	N/A	N/A	N/A	.01	N/A	.80
12 42 SS CONTAINER	CNTR M/T	N/A	N/A	N/A	N/A	N/A	.80	.50	N/A	N/A	N/A	.01	N/A	.80

JSCP STOW FACTORS AS A FRACTION OF USABLE CARGO SPACE (CONTINUED)

SHIP TYPE	SHIP COMPARTMENT STOW UNITS	CARGO CLASS 1 10 B/B GEN	2 A0 B/B AMMO	3 1V B/B OVSZ	4 2N NSDA	5 30 VEHICLES	6 C GEN 50 C VEH 5V REEFR 5R	7 5A CT AMMO	8 2F FLT CRFT	9 70 PAX	10 7M MEDVAC	11 7N NEO	12 80 POL	SHIP STOW FACTOR
13 43 SS CNTR-RO/RO	CNTR M/T RO/RO	N/A N/A	N/A N/A	N/A 1.1	N/A 1.1	N/A .75	.80 .75 *	.50 N/A	N/A N/A	N/A N/A	N/A N/A	.01 N/A	N/A N/A	1.15 .80
14 46 TACS-AUX CRANE	CNTR M/T	?	?	?	?	?	?	?	?	?	?	?	?	?
15 48 NSS MOD CNTR	CNTR M/T LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
16 49 SS MOD CNTR	CNTR M/T LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
17 51 PASSENGER	PAX	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.00	1.00	1.00	N/A	71.40
18 TANKER 61 SHALLOW DRAFT	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.00	1.00
19 TANKER 62 SMALL	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.00	1.00
20 TANKER 63 HANDY SIZE	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.00	1.00
21 TANKER 64 MEDIUM	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.00	1.00
22 TANKER 65 LARGE	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.00	1.00
23 TUG BARGE 68 LIQUID	CBBL	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.00	1.00
CARGO CLASS STOW FACTOR		1.00	.80	.70	.80	.67	1.00	.625	1.00	.014	.014	.014	1.00	

* CONTAINERS IN SQ FT COMPARTMENTS ARE STACKED TWO HIGH

CARGO CLASS/SHIP TYPE ASSIGNMENT PREFERENCE TABLE

TABLE ENTRIES REPRESENT THE DESIRABILITY
OF THE ASSIGNMENT OF A PARTICULAR CARGO
CLASS TO A PARTICULAR SHIP TYPE
1 IS MOST FAVORABLE

SHIP TYPE	SHIP COMPARTMENT STOW UNITS	CARGO CLASS											
		1 10 B/B GEN	2 A0 B/B AMMO	3 1V B/B OVSZ	4 2N NSDA	5 30 VEHICLES	6 C GEN 50 C VEH 5V REEFR 5R	7 5A CT AMMO	8 2F FLT CRFT	9 70 PAX	10 7M MEDVAC	11 7N NEO	12 80 POL
1 11 BREAKBULK	B/B M/T	1	1		3	3						2	
	DECK LO/LO	3		1	2	1	2		1				
2 12 SS CNTR-B/B	CNTR M/T						1	1					
	B/B M/T	1	1		3	3						2	
3 TUG BARGE DRY 13	B/B M/T	1	1				2					3	
4 14 LST	B/B M/T	2	2									3	
	LO/LO			2	3	2	3						
5 15 SEATRAIN	LO/LO	2	1	1	1	1	3	3	1			2	
6 21 RO/RO	RO/RO			1	1	1	3	3				2	
7 24 FAST SEALIFT	B/B M/T	?	?	?	?	?	?	?	?	?	?	?	?
	CNTR M/T	2	2			1						2	
	RO/RO			1	1	1	3						
	LO/LO	?	?	?	?	?	?	?	?	?	?	?	?
8 25 MARITIME PREP	RO/RO	?	?	?	?	?	?	?	?	?	?	?	?
	LO/LO	?	?	?	?	?	?	?	?	?	?	?	?
9 31 LASH	B/B M/T	1	1	1	1	2	2	2	1			2	
	CNTR M/T	?	?	?	?	?	?	?	?	?	?	?	?
10 32 SEABEE	B/B M/T	1	1	1	1	2	2	2	1			2	
	CNTR M/T	?	?	?	?	?	?	?	?	?	?	?	?
11 41 NSS CONTAINER	CNTR M/T						1	1				2	
12 42 SS CONTAINER	CNTR M/T						1	1				2	

(2/16/89, REVISED 1/23/90)

CARGO CLASS/SHIP TYPE ASSIGNMENT PREFERENCE TABLE (CONTINUED)

TABLE ENTRIES REPRESENT THE DESIRABILITY
OF THE ASSIGNMENT OF A PARTICULAR CARGO
CLASS TO A PARTICULAR SHIP TYPE
1 IS MOST FAVORABLE

SHIP TYPE	SHIP COMPARTMENT STOW UNITS	CARGO CLASS											
		1 10 B/B GEN	2 A0 B/B AMMO	3 1V B/B OVSZ	4 2N NSDA	5 30 VEHICLES	6 C GEN 50 C VEH 5V REEFR 5R	7 5A CT AMMO	8 2F FLT CRFT	9 70 PAX	10 7M MEDVAC	11 7N NEO	12 80 POL
13 43 SS CNTR-RO/RO	CNTR M/T RO/RO			1	1	1	1 3	1				2	
14 46 TACS-AUX CRANE	CNTR M/T	?	?	?	?	?	?	?	?	?	?	?	?
15 48 NSS MOD CNTR	CNTR M/T LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
16 49 SS MOD CNTR	CNTR M/T LO/LO	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?	? ?
17 51 PASSENGER	PAX									1	1	1	
18 TANKER 61 SHALLOW DRAFT	CBBL												1
19 TANKER 62 SMALL	CBBL												1
20 TANKER 63 HANDY SIZE	CBBL												1
21 TANKER 64 MEDIUM	CBBL												1
22 TANKER 65 LARGE	CBBL												1
23 TUG BARGE 68 LIQUID	CBBL												1

(2/16/89, REVISED 1/23/90)

APPENDIX F

COMMON VARIABLE DEFINITIONS

The organization of SAIL's data reflects the nature and use of the code itself: SAIL is a transient system called into use by the larger SEASTRAT system to make a set of computations on one case of data. It accepts information from SEASTRAT in the form of files, processes this information, and returns answers on files. The data inside of SAIL exist only for the duration of the computations.

SAIL's information is stored in about four dozen FORTRAN common blocks. Each of these blocks contains scalar variables and arrays that have some relationship to each other. These relationships may be of two types: data related to a particular entity or data used in a particular section of code. An example of an entity relationship is a common block named SHIP, which contains most of the ship-related data. Common block SCHED is an example of code-related data: it contains variables required by the routines of the Simulation Subsystem. Appendix I shows a mapping of common blocks to the FORTRAN codes. It can be seen in that map that certain basic data, such as ship and port data, are used in almost all sections of SAIL. More specialized commons appear in only a few places.

In the code, each common block begins with comments telling its purpose, and the variables that appear in the common are defined. This appendix organizes all of these definitions in alphabetical order and indicates in which block the variable resides.

Throughout this appendix, important subscripts are standardized.

S	refers to internal ship numbers
P	refers to internal port numbers
C	refers to internal cargo record numbers
CH	refers to channel numbers
TR	refers to the trip number of a ship
CL	refers to the cluster number (of a cluster of channels)
ROW	refers to the row number of the transportation problem
COL	refers to the column number of the transportation problem
CLS	refers to the cargo class subscript (B/B, B/B Ammo, Container, etc.)
ST	refers to the cargo stowage type (MT, square feet, Cont MT, PAX, CBBLS)
PT	refers to the problem type (dry cargo=1 or POL=2)
F	refers to the port facility subscript
SHPT	refers to the type of ship

All data elements are Integer*4 unless otherwise noted.

Variable Name	Description	Common
ACOEF(L)	a coefficient of risk vs speed for ocean link L where RISK=ACOEF/(SPEED**BCOEF)	OCEAN
ACORD(I)	I-th activity of a ship while simulating a trip during ship routing: 'L' = load 'U' = unload 'I' = intermediate stop (CHARACTER*1)	CYCLE
ACSEQ(S,I)	the I-th activity of a ship in simulation subsystem: 'L' = load 'U' = unload 'B' = bunker 'I' = intermediate stop (CHARACTER*1)	SCHED
ACSEQT(I)	temporary storage for ACSEQ(S,I) (CHARACTER*1)	PATHCOM
ACT1ST(I)	true if activity I is the first activity at a port for a ship (LOGICAL)	POSTCM
ACTOK(I)	true if activity I meets minimum port activity levels to warrant a port stop (LOGICAL)	SCHED
ACTPRP(S)	activation preparation days	SHIP
ACTTIM(I)	days duration of the I-th activity from LISTOUT (REAL)	POSTCM
ADDCH	the number of channels placed on ships	CYCLE
ADDS(S)	administrative delay days	SHIP
AFOEID(P)	incremental days ship is released from an AFOE port following delivery of cargo	PORT
AFOERD(P)	AFOE release date (C-days)	PORT
ALD(C)	available to load date	CARGO1
AMBAD(I)	the amount of the I-th channel on a ship being rejected in the ship reuse scheme	REUSE

Variable Name	Description	Common
AMLOAD(CH)	the amount of channel CH loaded at any time during the simulation	LIFCAP
AMSEQ(S,I)	the amount of channel CHSEQ to be carried at event I	SCHED
AMSEQT(I)	temporary storage of AMSEQ(S,I)	PATHCOM
AMSKED(S,TR,I)	the amount of cargo, in appropriate units, from the solution vector, X	SCHED
AMTAT(I,ST)	the amount of cargo on compartment ST of the target ship when it leaves the I-th port	REFCOM
AMTCL(S,CL)	amount of cargo in a cluster on ship S	CYCLEA
AMTON	the amount of cargo on a ship during trip evaluation	CYCLE
AMTR(I)	the amount of the I-th channel on the rejected ship that has not been trial loaded onto a free ship	REUSE
AMTREM(CH)	the amount remaining in channel CH; reset to CHREM at the beginning of evaluation of each candidate	NOPT
ARCDIM	dimension of the number of arcs in the generalized network code	PARETC
ARISK(PP,PHASE)	the A-coefficient of risk, multiplied by 1000 and stored in a 2-byte integer	DISCOM
ARP(S)	the arrival port of ship S	LOCATE
ARRIVE(S,TR,P)	arrival time of trip TR of a ship at a port (INTEGER*2)	SETUP
ASSOC(P)	political association of port P: 'N' = NATO port 'U' = United States port 'K' = Korean port 'E' = everywhere else (CHARACTER*1)	PORT
ATD(S)	the actual time of departure (in days) of ship S	LOCATE
ATSTAT	when true, attrition risk factors will be applied and risk statistics will be generated	PARETC
ATTGET	file from which interport attrition risks are read	PARETC

Variable Name	Description	Common
ATTRIT(S)	the day a ship was attrited	SHIP
ATTSAV	file name for saving interport attrition risks	PARETC
ATTZAP	when true, shipping resources will be attrited	PARETC
BARGE	draft (in feet) of barges for LASH and SEABEE ships	PARETC
BCOEF(L)	B-coefficient of risk for link L	OCEAN
BEGDIS(S,P)	the distance from a ship's original position to port P	SAD
BEST(S)	the amount of cargo in the best cluster for the ship	CYCLEA
BIGCST	a prohibitively high cost for use in optimization	PARETC
BNKER(S)	bunkering flag written on file LISTOUT (CHARACTER*1)	SCHED
BOOM(S)	maximum boom capacity in long tons	SHIP
BPRTAT(S)	going through the file backwards, this is the port number where the ship is loading	ARRDEP
BRISK(PP,PHASE)	the B-coefficient of risk, multiplied by 1000 and stored in a 2-byte integer	DISCOM
BUNK(S)	calculated travel time factor to account for bunkering (REAL)	SHIP
BUNKDE	bunker delay, in fractions of days (REAL)	SHIP
BUNKER	multiplier on miles used to allow for refueling time; used in ship routing only (REAL)	PARETC
CANCAS(PHASE)	the canal condition of the phase; 1=both canals open, 2=Panama closed, 3=Suez closed, 4=both closed	DISCOM
CANDAY(I)	start day of the I-th set of canal conditions	PARETC
CANDIM	dimension of the number of canal status changes	PARETC

Variable Name	Description	Common
CAP(S,ST)	capacity in appropriate units for the compartment 1 = MT 2 = square feet 3 = container MT 4 = passengers 5 = CBBLS	SHIP
CAPFRE(ST)	the remaining capacity of the ST-th compartment of free ship	REUSE
CAPR(ST)	temporary remaining space in compartment ST	NOPT
CAPREM(S,ST)	running total of capacity remaining in compartment ST of ship S	POSTCM
CAPS(S)	total capacity of a ship, all compartments, with a nominal adjustment for stow factor	CYCLE
CAPTOT(S)	ship capacity as a function of all stow types	SHIP
CAROS(S,I)	the I-th cargo number loaded on a ship	SCHED
CARTOT(PT)	total quantity of cargo in problem type PT; for dry cargo, this adds M-tons and PAX	PARETC
CDIM	cargo dimension	PARETC
CEXCL(C)	the amount (in the measure of CQTY1) of cargo C that has been eliminated from the problem, generally for being too late	CARGO1
CFAC(C)	preferred port type for a cargo	CARGO2
CGOCH(C)	the number of the channel to which a cargo is assigned	CARGO1
CHALIS(I)	the actual channel number of the I-th item in the list	CHVAR
CHAMMO(CH)	TRUE if a channel is ammunition (LOGICAL)	CHANCOM
CHAMT1(CH)	amount of cargo in a channel in MT, PAX, or CBBLS	CHANCOM
CHAMT2(CH)	amount of cargo in a channel in square feet	CHANCOM
CHANGE(ST)	amount of change to the stow-adjusted cargo in compartment ST during refinement (REAL)	REFCOM

Variable Name	Description	Common
CHAR(I,CC)	temporary array for reading port types from file FACIDAT, I=1 is primary; I=2 is secondary	INPUT
CHBAD(I)	the I-th channel on the bad ship	REUSE
CHBEST(S)	a channel in the best cluster of ship S	CYCLEA
CHCGO(C)	list of cargo numbers sorted by the channels to which the cargos are assigned	SCHED
CHCL(COL)	the I-th channel clustered with the prime channel	NOPT
CHCLAS(CH)	cargo class of the channel	CHANCOM
CHCLUS(CH)	the cluster number of channel CH	CHANCOM
CHCOL(CH)	the column number of a channel, used in refining	REFCOM
CHCOLM(CH)	the column number of a channel, used in routing	CYCLE
CHDELY(CH)	the channel's days delay at intermediate port; the maximum of delays of all cargos in the channel	CHANCOM
CHDEP(CH)	relates dependent and independent channels; if positive, channel CH is a dependent channel and CHDEP contains the associated independent channel number; if negative, CH is an independent channel and CHDEP is the negative of the dependent channel number	CHANCOM
CHDIM	channel dimension	PARETC
CHDIST(CH)	the distance a channel will travel without diverting for other channels; includes bunker factor	CHANCOM
CHDUM(CH)	amount of dependent channel CH that is scheduled on a ship but is dependent on a channel that is not scheduled	SCHED
CHEAD(CH)	the earliest EAD of cargos in the channel	CHANCOM
CHFAC(CH)	port facility type preferred for the channel	CHANCOM
CHFINE(I)	TRUE when service to the I-th channel in a ship path is completed (LOGICAL)	SCHED
CHGONE(CH)	the amount of channel CH that has been shipped on a real ship; used in DUMSHP	REFCOM

Variable Name	Description	Common
CHINTR(I)	TRUE if there is no short intermediate stop for the I-th activity or if the stop has been completed (LOGICAL)	SCHED
CHLADA(CH)	the earliest LAD of cargos in the channel	CHANCOM
CHLD(I)	load time of the I-th channel of the channel list--see CHLIST(I) (REAL)	CYCLE
CHLIST(I)	the I-th channel in a list of channels in the ship trip	CYCLE
CHLOAD(CH)	the amount of a channel (in MT, PAX, or CBBLS) that is loaded	SCHED
CHMT(I)	the amount of the I-th channel in the channel list--see CHLIST(I)	CYCLE
CHMXLT(CH)	the maximum number of days of lateness for a channel	CHANCOM
CHNUSE(CH)	true if channel CH is to be used and should be reported (lift capability computations only)	LCCOM
CHOK(CH)	TRUE if channel CH can be assigned (LOGICAL)	CHOKCOM
CHPLAN(CH)	OPLAN number for channel CH	CHANCOM
CHPNT1(CH)	first cargo in CHCGO that is in channel CH	SCHED
CHPNT2(CH)	last cargo in CHCGO that is in channel CH	SCHED
CHPNTT(CH)	set to CHPNT1 initially; used in building CHCGO	SKED
CHPOD(CH)	channel's POD	CHANCOM
CHPOE(CH)	channel's POE	CHANCOM
CHPOI(CH)	channel's POI	CHANCOM
CHREJ(CH)	a counter for the number of times channel CH has been rejected as a prime channel due to lateness; after several tries, the channel is no longer allowed to be a prime	CYCLEA
CHREM(CH)	the amount of a channel remaining to be routed	NOPT

Variable Name	Description	Common
CHRQCN(CH)	indicates special features of a channel: 'A' = AFOE 'C' = CONSOL ' ' = no special feature (CHARACTER*1)	CHANCOM
CHSDIM	dimension of the number of channels on a single ship	PARETC
CHSEQ(S,I)	the I-th channel event of a ship, as determined by subroutine PATH; there are two events for each channel: load and unload (INTEGER*2)	SCHED
CHSEQT(I)	temporary storage of CHSEQ(S,J)	PATHCOM
CHSHIP(CH)	preferred ship for a channel	CHANCOM
CHSHPT(CH)	preferred ship type for a channel	CHANCOM
CHSKED(S,TR,I)	the I-th channel scheduled for the TR-th trip of a ship as found in the solution vector (INTEGER*2)	SCHED
CHSORT(I)	this array contains the I-th channel number when channels are sorted by cargo class	CHANCOM
CHST(I)	the stow type used by the I-th channel--see CHLIST	CYCLE
CHSTRT(I)	TRUE when service to a channel has begun in subroutine PATH (LOGICAL)	SCHED
CHT0(CH)	C-day of the latest available to load date for all cargos in the channel	CHANCOM
CHT1(CH)	the beginning day of channel CH's time interval	CHANCOM
CHT2(CH)	the ending day of channel CH's time interval	CHANCOM
CHTYP(CH)	stowage type of the channel	CHANCOM
CHUAL(CH)	number of days of unavoidable lateness in the channel	CHANCOM
CHUN(CH)	unload time of the I-th channel in the list CHLIST (REAL)	CYCLE
CHUNIT(CH)	military unit of the channel; blank for non-unit (CHARACTER*6)	CHANCOM

Variable Name	Description	Common
CID(I)	cargo identification of the I-th record from LISTOUT	FINI
CLAMT(I)	the amount of cargo remaining in the I-th channel	NOPT
CLATE(C)	TRUE if the cargo is going to be delivered late (LOGICAL)	CARGO1
CLBASE(CL)	the base channel of cluster CL	CHVAR
CLOSE	closure date of all activities	SCHED
CLUSTR(I)	the I-th cluster on the target ship	REFCOM
CLUTOT(CL)	the total amount of cargo in cluster CL, decremented in NOPT	CHANCOM
CNOW	day planning occurs; if positive, it indicates reflow	PARETC
COEF(L,2)	these are linear equation coefficients of line L (REAL)	OCEAN
COLCL(I)	the column associated with the I-th clustered channel	NOPT
COLCNT(ROW)	the number of columns with solutions in them in a row	VECTOR
COLDIM	dimension of the number of columns in the transportation problem	PARETC
COLDON(COL)	during the routing of a ship, this contains the quantity of the channel represented by column COL (of the transportation problem) that is on the ship	CYCLE
COLGO(COL)	used in subroutine ENRICH; indicates a column was used on a ship (LOGICAL)	CYCLE
COLJ(J)	the column number in the transportation problem corresponding to the J-th variable in the general network code	SETUP
COLM(I,ROW)	the column number of the I-th candidate for an entering variable in a row (INTEGER*2)	VECTOR
COLMRK(COL)	TRUE if column COL is on a sailed ship being analyzed in subroutine SAILUM (LOGICAL)	CYCLE

Variable Name	Description	Common
COLORD(I)	the I-th entry of this array contains the solution vector index (the index of the X vector) of the transportation problem sorted by columns	TBASIS
COLOS(I)	the column number of the I-th channel on a ship	CYCLEA
COLST(COL)	the first index of vector COLORD in column COL	TBASIS
COLTOT(COL)	the last index of vector COLORD in column COL	TBASIS
COMP(CH)	TRUE if a channel will be removed--compressed--from a set of channels (LOGICAL)	CHVAR
COMPLX(P)	port complex GEOLOC code of a port (CHARACTER*5)	PORT
COMPST(I)	the measure of the I-th compartment	STOW
COMPT(I)	compartment type number of the I-th LISTOUT record	FINI
CONNR(COL)	the number of the channel constraint for the channel in column COL of the transportation problem	SETUP
CONSDR	the number of secondary channel loadings considered by subroutine CHEVAL	PARETC
COSDIM	dimension of cargos on a single trip of a ship	PARETC
COST(COL,ROW)	the cost matrix	COSTCOM
COSTV(ROW)	contains the costs of a particular column of the cost matrix	CYCLEA
COUNT(CH)	the number of cargos in each channel	SKED
CPLAN(I)	the plan id of the I-th LISTOUT record	FINI
CPOD(C)	the POD of the cargo	CARGO1
CPOE(C)	the POE of the cargo	CARGO2
CPOI(C)	the POI of the cargo	CARGO1

Variable Name	Description	Common
CPOINT(C)	cargo with intermediate stops is split into two cargos; the first leg of the trip forms the independent cargo of the pair and retains the original cargo's number; a new cargo is formed for the second leg, and it is called the dependent cargo (a new number is created for it); for independent cargos, CPOINT contains the negative of the dependent cargo number; for dependent cargos, it contains the independent cargo number	CARGO1
CPORT(I)	port name of the I-th LISTOUT record	FINI
CRISK(I)	cargo risk of I-th record; on LISTOUT file, this is the probability that the cargo survives	POSTCM
CQTY1(C)	cargo quantity in MT, PAX, or CBBLS	CARGO1
CQTY2(C)	cargo quantity in square feet	CARGO1
CSAMT(S,I)	the amount of the I-th cargo on a ship	SCHED
CSHIP(I)	ship's NISC number of the I-th LISTOUT record	FINI
CSSTO(S,I)	the ship stowage used by the I-th cargo on a ship (INTEGER*2)	SCHED
CSTOW(C)	the type of stowage preferred for a cargo	CARGO1
CSXTR(S,I)	if the I-th cargo on a ship is loaded by the simulation subsystem to an amount greater than indicated in the solution vector, this contains the quantity of that extra load	SCHED
CTDIM	cargo type dimension	PARETC
CTOGO(C)	cargo to go; the quantity of cargo C remaining to be loaded	SCHED
CTRAN(I)	ship transit time through canal I (REAL)	PARETC
CTRIP(I)	ship's trip number of the I-th LISTOUT record	FINI

Variable Name	Description	Common
CTYPE(C)	cargo type: 1 = '10', B/B 2 = 'A0', B/B AMMO 3 = '1V', B/B OVERSIZE 4 = '2N', NSDA 5 = '30', VEHICLES 6 = '50', '5R', '5V', CONTAINERS 7 = '5A', CONTAINER AMMO 8 = '2F', FLOATING CRAFT 9 = '70', PAX AND MEDIVAC 10 = '7N', NEO 11 = '80', POL	CARGO1
D1(I)	ship's arrival date in port of the I-th LISTOUT record	FINI
D1J(I)	distance from port I to port J on a ship's route; used in bunker calculations	BUNK
D2(I)	ship's departure date from port of the I-th LISTOUT record	FINI
DATE(I)	date of the I-th record in the LISTOUT file (REAL)	ARRDEP
DATE2(I)	the date of the I-th record in LISTOUT if slack is removed to avoid earliness (REAL)	ARRDEP
DATE3(I)	the date of the I-th record in LISTOUT if slack is removed as much as possible without causing lateness (REAL)	ARRDEP
DBARKD(CLA, SHPT,ST)	the amount of cargo class CLA unloaded from compartment ST of ships of type SHPT	SCHED
DDELAY(C)	days delay at intermediate port	CARGO1
DEFCAP(SHPT)	default fuel capacity for ships of type SHPT in long tons	PARETC
DEFUSE(SHPT)	default fuel use for ships of type SHPT in long tons per day	PARETC
DEL1(S)	the number of days a ship may be delayed to remove schedule slack up to the EAD	ARRDEP
DEL2(S)	the number of days a ship may be delayed to remove schedule slack up to the LAD	ARRDEP

Variable Name	Description	Common
DEMAND(COL,PT)	total demand of column COL, problem type PT	TRANSP
DEP(S)	the departure port of ship S	LOCATE
DEPART(S,TR,P)	departure time of ship S, trip TR, at port P (INTEGER*2)	SETUP
DEPDEL(I)	if loading, this is the port departure time; if unloading or intermediate, it is the delivery time (REAL)	ARRDEP
DEPOTE(I)	port departure day built by activity	POSTCM
DIJ(I)	distance from port I to a specified port, J, in the list of ports on a ship's route; used in bunker computations	BUNK
DISDIM	dimension of distance vector, PPDIST	PARETC
DISGET	file name from which port-to-port distances are to be read (CHARACTER*7)	PARETC
DISSAV	file name for saving port-to-port distances (CHARACTER*7)	PARETC
DISSUM(CH)	distance from POE of prime channel to POE of channel CH plus the distance from POD of prime channel to POD of channel CH; computed in subroutine SECON1	CYCLE
DIVDAT	date of changeover to second interval width; must be a multiple of INT1	PARETC
DOBUNK	TRUE if bunker computations are to be made (LOGICAL)	PARETC
DOKTIM	time to dock a ship, in fractions of a day (REAL)	PARETC
DOREEF	TRUE if load refinement is to be made in the simulation subsystem (LOGICAL)	PARETC
DORESD	true if subrouting SIMOVR determines reseeding is to be done	REUSES
EAD(C)	early arrival date of a cargo	CARGO1

Variable Name	Description	Common
ENDS(I,4)	latitude and longitude of each end of line I (REAL)	OCEAN
ERCODE(I)	code indicating the reason the I-th record was excluded	POSTCM
EVDIM	dimension of events (loads and unloads) of channels on a ship	PARETC
ETA(S)	the estimated time of arrival (in days) of ship S	LOCATE
EWD	overall early warning days	PARETC
EWDS(S)	early warning days, by ship	SHIP
EXTOT(ST)	total cargo excluded by stow type	PARETC
EXTRA(S,TR,ST)	unused capacity on ship S, trip TR, stowage type ST expressed in MT, SQ FT, PAX, or CBBLS, as appropriate	SCHED
FAC(I)	map of IFAC(I) to characters (CHARACTER*1)	POSTCM
FACAT(S)	the current port facility type of ship S	SCHED
FACTOR(CL,E/L)	the weighing factor for each cargo class, depending on whether the cargo is early or late; used to find a measure of the solution quality (REAL)	POSTCM
FACUSE(CL,SHPT, ST,I)	the I-th (I = 1 or 2) port facility type that may be used for loading cargo class CL onto stowage type ST of ship type SHPT	STOW
FAML	index of first ammunition load port in a ship's route in the ship routing subsystem	CYCLE
FAMUL	index of first ammunition unload port in a ship's route in the ship routing subsystem	CYCLE
FAS(CLAS,SHPT, ST)	favorability of placing cargo class 'CLAS' in compartment 'ST' of ship type 'SHPT': 1=best; 2=next best; 3=ok; 4=least favorable or disallowed	STOW
FCSEQ(S,I)	facility type of the I-th activity of ship S (INTEGER*2)	SCHED
FCSEQT(I)	temporary storage of FCSEQ(S,J)	PATHCOM

Variable Name	Description	Common
FDIM	dimension of port facility types	PARETC
FEWS(CLASS)	amount of cargo CLASS for which there are few acceptable ships	REPORT
FIFO(P)	a small increment of time added to the TNEXT of a ship that is queued at a dock; maintains a first-in:first-out queuing policy (REAL)	SCHED
FINISH(CH)	time a channel finishes unloading as a ship trip is being evaluated	CYCLE
FITPAN(S)	TRUE if a ship can go through the Panama Canal (LOGICAL)	SHIP
FITSUE(S)	TRUE if a ship can go through the Suez Canal (LOGICAL)	SHIP
FLEET(S)	merchant fleet code	SHIP
FLTDIM	fleet dimension	PARETC
FOB(S)	fuel on board ship S, in long tons	BUNK
FORBID(P)	true if port P is a forbidden port to special ships	POXCOM
FREES(J,PT)	the J-th free ship of problem type PT	REUSES
FTDIM	dimension of ship fuel types	PARETC
FUEL(S)	type of fuel used by a ship (CHARACTER*3)	SHIP
FUELCP(S)	fuel capacity for a ship's own use, in long tons	SHIP
FUELTP(C)	cargo fuel type (CHARACTER*3)	CARGO2
FUELUS(S)	a ship's fuel consumption in long tons per day	SHIP
FUL	index of first port at which any type of cargo is unloaded in the ship routing subsystem	CYCLE
GNODIM	dimension for the number of nodes in the generalized network code	PARETC

Variable Name	Description	Common
GOODNS	a measure of the quality of the solution; a function of total lateness and earliness (REAL)	POSTCM
I1(S)	index number in listout of first use of ship S	RLCOM
IACT(I)	activity of the I-th record of the LISTOUT file: 'U' = unload ' ' = not unload (CHARACTER*1)	ARRDEP
IAMT(I)	the amount of cargo in record I	POSTCM
IBEST(S)	the solution vector index number of the best channel for ship	CYCLEA
IC(I)	the cargo number of the I-th record of the LISTOUT file	ARRDEP
IDMIN	if days delay at intermediate is not greater than IDMIN, the ship pauses but does not unload	PARETC
IFAC(I)	the port facility type used: 1 = A, ammunition 2 = B, container ammunition 3 = C, container 4 = G, general 5 = P, petroleum	POSTCM
ILAD(I)	the late arrival date of the I-th record (for delivery records only) of the LISTOUT file	ARRDEP
ILATE(I)	lateness or earliness in the I-th record of the LISTOUT file	ARRDEP
ILATE2(I)	the amount of earliness after removing slack to the EAD in the I-th record of the LISTOUT file	ARRDEP
ILATE3(I)	the amount of earliness (or lateness) after removing slack to the LAD in the I-th record of the LISTOUT file	ARRDEP
INCOL(COL)	equals the column number that contains the cost to be used for the variable in column COL (for a given row)...contains zero if the column is not to be used in the general network code	SETUP

Variable Name	Description	Common
INDCH(COL,PT)	the channel number of a column in problem PT (INTEGER*2)	TRANSP
INDEX1(I)	the record number of the first link for node I	DCALC
INDEX2(I)	the record number of the last link for node I	DCALC
INDS(ROW,PT)	the ship number corresponding to a row of the transportation problem, problem PT (INTEGER*2)	TRANSP
INDST(ROW,PT)	the stow type of the ship corresponding to a row of the transportation problem, problem PT (INTEGER*2)	TRANSP
INDTR(ROW,PT)	the trip number of the ship corresponding to a row of the transportation problem, problem PT (INTEGER*2)	TRANSP
INFOB	assumed initial fraction of fuel on board ships (REAL)	PARETC
INSEQ(S,I)	index of the solution vector for the I-th channel assignment to a ship (INTEGER*2)	SCHED
INSEQT(I)	temporary storage of INSEQ(S,I)	PATHCOM
INSKED(S,TR,I)	index of the solution vector of the I-th channel assignment on ship S, trip TR (INTEGER*2)	SCHED
INT1	length of the early channel intervals, days	PARETC
INT2	length of the later channel intervals, days	PARETC
INTDIM	time interval dimension	PARETC
INXOS(S,I)	the solution vector index of the I-th cargo on a ship (INTEGER*2)	SCHED
IPNT(I)	initially, this is a pointer from dependent leg records of the LISTOUT file to independent leg records; for use during removal of schedule slack, IPNT contains these pointers with their signs reversed	ARRDEP

Variable Name	Description	Common
IPOINT(C)	for dependent cargo legs, points to the record number of the associated independent leg; for use in the reporting subsystem	SCHED
IQSHIP(I)	queue time of the ship on the I-th LISTOUT record at a port	POSTCM
IREM(ROW,I)	contains the solution vector index: I = 1, of a column of interest I = 2, of the dummy column	CYCLEA
ISEQ(I)	sequence number of the I-th record of LISTOUT	POSTCM
ISHP(I)	ship number of the I-th record of LISTOUT file	ARRDEP
ISPEED(S)	inverse of speed, days/mile	SHIP
ISTORE	the number of solutions placed on sailed ships	CYCLE
ISTYPE(I)	the stow type of the I-th record of LISTOUT file	POSTCM
ITER	total iteration count of the transportation problem	TRANSP
ITMXTP	maximum iterations allowed in the transportation problem; if exceeded, the current solution will be used	PARETC
ITR(I)	the ship trip number of the I-th record of LISTOUT file	ARRDEP
J1(I)	first event index of the I-th port call of a ship	PATHCOM
JA(S,TR)	the number of channel events for ship S, trip TR	SCHED
JAYS(I)	during bunker computations, this contains the activity index number for the I-th port on the ship's route	BUNK
JL(I)	last event index of the I-th port call of a ship	PATHCOM
JLAST(S)	the last event index of the sequence arrays (CHSEQ, AMSEQ, etc.) that a ship can reach because of fuel restrictions	BUNK
JMAX(S)	number of event indexes for a ship in arrays CHSEQ, AMSEQ, ACSEQ, etc.	SCHED
JNEXT(S)	the next event index for a ship in arrays CHSEQ, etc.	SCHED

Variable Name	Description	Common
KDIM	dimension of the number of ships active at a port facility at one time	PARETC
KEPTIT	the number of secondary channel loadings accepted	PARETC
KMAX(P,F)	the number of intervals for port P, facility F	SCHED
KOREA	TRUE if Korean ships are limited to Korean ports (LOGICAL)	PARETC
LAD(C)	a cargo's latest arrival date	CARGO1
LAML	index of last ammunition load port in a ship's route in the ship routing subsystem	CYCLE
LAMUL	index of last ammunition unload port in a ship's route in the ship routing subsystem	CYCLE
LASTCG(S)	set to true in DEBARK if any cargo unloaded from ship S is AFOE (LOGICAL)	SCHED
LASTPT(S)	the last port visited by ship S on previous trip	CYCLE
LASTSP	the last ship processed by PORTTM, used in port throughput usage updates	SCHED
LASTTM(S)	the time ship S will be ready to leave port LASTPT(S)	CYCLE
LATENS	quality of the schedule based on lateness only (REAL)	POSTCM
LCAP(ST,T)	total amount of lift capability of stow ST in time T (lift capability computations only)	LCCOM
LCAPP(ST,T,CH)	amount of lift capability of stow ST, time T, in channel CH (lift capability computations only)	LCCOM
LCCUT	cutoff date for lift capability computations	PARETC
LDMIN1(SHPT)	minimum fraction load for first 30 days for a ship type (REAL)	PARETC
LDMIN2(SHPT)	minimum fraction load for next 30 days for a ship type (REAL)	PARETC
LDMIN3(SHPT)	minimum fraction load for remainder of schedule (REAL)	PARETC

Variable Name	Description	Common
LDRATE(CL, SHPT,ST)	the loading rate in units per day for a cargo class, ship type, and stow type; the units are MT for all cargo other than PAX or POL	STOW
LEAVE(S,TR,P)	the time trip TR of ship S leaves port P	BUNK
LENGTH(I)	distance from the source node to node I in shortest path calculations	DCALC
LINDIM	dimension on number of lines	PARETC
LINKID(I)	the identification of link I (CHARACTER*8)	DCALC
LINKSV(2L)	link number in the forward star routine	DCALC
LL	index of the last load port on a ship's route	CYCLE
LMAX	number of channels on the rejected ship	REUSE
LNKDIM	dimension on number of links	PARETC
LNR	the number of elements in PNTR for any one sifted group	CHVAR
LOAD	default time to load a ship, in days (REAL)	PARETC
LOADED(ST)	keeps track of the load on a ship for stow type ST; used to check for overloading	POSTCM
LOCSHP(S)	true if ship S is in the LOCADAT file	LOCATE
LOGROW(ST)	keeps track of compartments that need to be represented in the solution; initially true for all compartments; set false as they are included in the solution (LOGICAL)	CYCLE
LUL(I)	load/unload indicator for record I of the LISTOUT file	FINI
MARK(I)	TRUE if the I-th entry in the solution vector has been evaluated on an earlier iteration (LOGICAL)	CYCLE

Variable Name	Description	Common
MARK2(I)	similar to MARK, but applies only to the loop in the prime routines (APRIME or NPRIME) that contains the call to CHEVAL; it is used to permit rejected channels a second chance (LOGICAL)	CYCLEA
MARKA(CH)	TRUE when channel CH has had a chance to be routed and deserves no further chance (LOGICAL)	NOPT
MARKER(S)	TRUE if a ship has been, or should be, refined (LOGICAL)	REFCOM
MARKIT(I)	used in building POINTR (LOGICAL)	SCHED
MARKR(I)	marks the remaining capacity queuing arrays to be kept after compression (LOGICAL)	SCHED
MARKX(I)	TRUE if the I-th element of the solution vector has been fully accounted for (LOGICAL)	REFCOM
MAXCAP(P,F)	maximum port throughput capacity of port P, facility F for any period	INPUT
MAXEAR	maximum days a cargo is permitted to be early	PARETC
MAXRAT(SHPT)	the maximum demand on port throughput of a ship of type SHPT	SCHED
MCAP(P,F)	military wartime throughput capacity of port P, facility F, in measurement tons	PORT
MCAPT(P,F,I)	a copy of MCAP(P,F) for I intervals (REAL)	SCHED
MILES(I)	the length of link I in nautical miles	OCEAN
MILESP	the length of the Panama link; equal to MILES(1)	OCEAN
MILESS	the length of the Suez link; equal to MILES(2)	OCEAN
MISPAT(S,P,F)	TRUE if ship S cannot use port P, facility F (LOGICAL)	MISCOM

Variable Name	Description	Common
MIX	TRUE if ammo and non-ammo cargo can be mixed on a ship (LOGICAL)	PARETC
MKEEP	the current value of the mileage evaluation criterion	CYCLE
MNFUEL	fraction of full fuel load at which refueling is required (REAL)	SHIP
MSPAN(2L)	the mileage of a link	DCALC
MULTY(L)	the function of length for a link	OCEAN
MXLTPX	maximum days lateness for replacements	PARETC
MXLTRA	maximum days lateness for resupply ammo	PARETC
MXLTRO	maximum days lateness for resupply other	PARETC
MXLTRP	maximum days lateness for resupply POL	PARETC
MXLTUN	maximum days lateness for unit cargo	PARETC
NATO	TRUE if NATO ships are limited to NATO PODs (LOGICAL)	PARETC
ND(PT)	number of columns in the transportation problem, problem PT	TRANSP
NEWNOD(N)	the new node number of old node N, used when eliminating nodes to reset PORTND(P), the node established for port P	DCALC
NID1(I)	the name of the first node for link I (CHARACTER*12)	DCALC
NID2(I)	the name of the second node for link I (CHARACTER*12)	DCALC
NLAT(I)	latitude of node I, in degrees and fractions; south is negative (REAL)	OCEAN
NLONG(I)	longitude of node I, in degrees and fractions; west is negative (REAL)	OCEAN

Variable Name	Description	Common
NLST(SHPT)	nominal load and unload rate by ship type	SHIP
NMLDUN(F)	nominal load and unload rate of ships at each facility type	STOW
NOCEAN	number of permanent ocean nodes	OCEAN
NODA(N)	a list of nodes between ports computed from array PRENOD and written into the route files	DCALC
NODDIM	dimension for the number of nodes	PARETC
NODE1(I)	one node number of link I	OCEAN
NODE2(I)	the other node number of link I	OCEAN
NODEID(I)	the name of node I (CHARACTER*12)	OCEAN
NODOUT(N)	if true, the node is to be eliminated due to ocean avoidance	DCALC
NOMQUE(P,F,I)	the nominal queuing delay (in days) at port P, facility F, time interval I (REAL)	QUEUE
NOWAIT(I)	initialized to FALSE in the simulation subsystem; it is set to TRUE when ships should not wait on the I-th solution vector index in the next pass through the simulation (LOGICAL)	SCHED
NR1	number of intervals of length INT1	PARETC
NRBOAT(P,SHP,T)	number of nominal ships of type SHPT needed at port P during time interval T (10 day intervals)	RLCOM
NRCGO	number of cargos	CARGO1
NRCH	number of channels on the current trip of a ship	CYCLE
NRCHNL	number of channels	CHANCOM
NRCL	number of clusters	CHANCOM
NRCLAS(CL)	number of channels of each cargo class	CHVAR
NREXCL	number of records on EXCLOUT file	PARETC

Variable Name	Description	Common
NRFAZE	the number of time phases	DISCOM
NRFREE(PT)	the number of free ships, problem type PT	REUSES
NRI	number of records on the LISTOUT file	FINI
NRINT	number of time intervals	CHANCOM
NRLATE(PT)	the number of bad trips, problem type PT	REUSES
NRLINE	number of lines	OCEAN
NRLINK	number of links	OCEAN
NRLIST	number of records on LISTOUT file	PARETC
NRNODE	number of nodes	OCEAN
NROS	number of channels on a ship	CYCLEA
NRP	number of activities on a ship's trip in routing	CYCLE
NRPLAN	the number of OPLANs included	PARETC
NRPO	number of ports in the ports list in bunker computations	BUNK
NRPORT	number of ports	PORT
NRRECS	number of records written	SCHED
NRSHIP	number of ships	SHIP
NRTRIP(S)	number of trips for a ship	SHIP
NS(PT)	number of rows (supplies) in problem PT	TRANSP
NUMONS(S)	number of cargos on ship S	SCHED
NXOS(S,IX)	pointer array to the on-ship vectors	SCHED
NXSEQ(S,JJ)	the index in the sequence vector of the JJ-th channel event of ship S for the sequence vectors	SCHED
ODRAFT(P)	feet of overdraft for port P	PORT
OFFSET(PL)	number of days plan PL is offset from the scenario day	PARETC

Variable Name	Description	Common
OGEO(C)	origin GEOLOC code (CHARACTER*4)	CARGO2
OK(I)	for a list of channels in a cluster, OK(I) is TRUE if the I-th channel in the list has adequate port activity to be loaded (LOGICAL)	CHVAR
OLDNEW(CH)	a pointer from an old list of channels to a compressed list of channels; used in subroutine CULL	CHVAR
OPLAN(PL)	the OPLAN name of the PL-th plan	PARETC
ORDLAT(I,PT)	average days of lateness on the I-th worst ship-trip for problem type PT	REUSES
ORDS(I,PT)	the I-th worst ship of problem type PT	REUSES
ORDTR(I,PT)	the trip number of the I-th worst ship, problem type PT	REUSES
OSMAX	the number of used elements in the on-ship vectors	SCHED
P3(I)	temporary array of port numbers	ARRDEP
P8(F)..P14(F)	temporary arrays for reading port data	INPUT
PAD(P)	port availability date for port P	PORT
PANB	maximum beam in Panama Canal	PARETC
PAND	maximum draft in Panama Canal	PARETC
PANL	maximum length of ship in Panama Canal	PARETC
PANPAS(S)	true if a ship has passed through the Panama Canal on the way to its present location	LOCATE
PANPTR(P,I)	first and last records for the route from port P to Panama	PTR
PANSUE(I)	I=1: first record for the Panama to Suez route; I=2: last record for Panama to Suez	PTR
PANX(S)	true if ship S used (or expects to use) the Panama Canal	LOCATE

Variable Name	Description	Common
PAR1	weight given to empty space when computing a ship's score; used to find the best ship for a cluster in heuristic routing (REAL)	PARETC
PAR2	tolerance on cost for finding the best ship to carry a channel in the heuristic routing process	PARETC
PAR3	tolerance on cost for finding ships almost as good as the best	PARETC
PAR4	days improvement required to refine a load; used in subsystem refine	PARETC
PAR5	range around the current minimum distance for accepting a channel in subroutine NXTBST; used in heuristic routing	PARETC
PAR6	number of days an ALD for a channel can be greater than the prime channel's ALD and still be accepted by subroutine NXTBST; heuristic routing	PARETC
PAR7	a multiplier (>1.0) that determines how much a ship's route can be extended to pick up a secondary channel (REAL)	PARETC
PAR8	the maximum number of days a secondary channel's estimated sailing date (CHT1) can exceed the primary channel's date	PARETC
PAR9	maximum miles between ports for channels to be clustered	PARETC
PAR10	this is a time window, in days, for determining if there is enough activity at a port to merit a ship stopping	PARETC
PAR11	the minimum quantity of cargo that a ship will stop to load	PARETC
PAR12	the minimum quantity of cargo that a ship will stop to unload	PARETC
PAR13	cost of the dummy row (except in the dummy column)	PARETC
PAR14	multiplier on minimum loads when eliminating ships prior to the ship loading subsystem (REAL)	PARETC

Variable Name	Description	Common
PAR15	miles a ship will go to pick up evacuees (NEO) as a prime cargo	PARETC
PAR16	the decision to place a secondary channel on a ship depends on how many alternative ships that channel may potentially go on; this parameter defines 'many' alternatives	PARETC
PAR17	similar to PAR16, this parameter defines 'some' alternatives	PARETC
PAR18	a multiplier on the criteria for accepting a secondary channel used in subroutine ACCEPT (REAL)	PARETC
PAR19	additional days to be added to the transit time of an alternative ship for a secondary channel to account for interactions among channels on that ship; used in routing	PARETC
PAR20	cost plateau for unsailed ships; separates sailed and unsailed ships in the costs of the transportation problem	PARETC
PAR21	days penalty for sheathing a ship for B/B ammo; used in simulation subsystem (REAL)	PARETC
PAR22	multiplier on the maximum days of lateness for a channel (CHMXLT) for setting cost to BIGCST or rejecting a prime channel; used in ship routing (REAL)	PARETC
PBEAM(P,F)	maximum ship beam allowed in port P, facility F	PORT
PBUNK(P)	TRUE if bunkering can be accomplished in this port (LOGICAL)	PORT
PC(I,PT)	the column number of the I-th entry in the solution vector of problem PT	TRANSP
PCHAN(I)	in the ship routing subsystem, the I-th activity of the ship is involved with the PCHAN(I)-th channel that is on the ship; the value in PCHAN thus becomes the subscript of the array CHLIST	CYCLE
PCNEW(I)	in the refinement process, this becomes the new PC vector for one problem type (PT)	REFCOM

Variable Name	Description	Common
PCSTOR(I)	temporary storage of the PC(I,PT) vector for the sailed ship	CYCLE
PCT(I)	temporary PC vector	SETUP
PDEP(S)	port departure time during backward scan; used in removing slack from the schedule	POSTCM
PDIM	dimension of number of ports	PARETC
PDIS(I)	distance from the port of activity I to the port of activity I+1	CYCLE
PDRAFT(P,F)	maximum draft of port P, facility F	PORT
PID(P)	GEOLOC code of a port (CHARACTER*4)	PORT
PLAN(C)	the plan number (subscript) assigned to this cargo C's plan	CARGO2
PLAST(S)	the last port a ship visited on the preceding trip	SAD
PLAT(P)	latitude of port P (REAL)	PORT
PLONG(P)	longitude of port P (REAL)	PORT
PLTH(P,F)	maximum ship length allowed in port P, facility F	PORT
PODX(CH)	POD of channel CH (lift capability computations only)	LCCOM
POEX(CH)	POD of channel CH (lift capability computations only)	LCCOM
POINTR(I)	the I-th cargo number on a ship, sorted by LAD	SCHED
POOL(I)	list of nodes with tentative lengths assigned	DCALC
PORDER(I)	the port number of the I-th activity in a string of ports to be visited by a ship on a single trip	CYCLE
PORTAT(S)	the current port location of ship S	SCHED
PORTL(I)	the I-th port visited by the target ship	REFCOM

Variable Name	Description	Common
PORTND(P)	node number of the port P	OCEAN
PORTS(I)	a list of ports on a ship's route between bunker ports; used in bunker computations	BUNK
POSTPN(S)	the number of days a ship is delayed waiting for an independent cargo to arrive	SCHED
POX	the port number of the transfer port	POXCOM
POXNAM	the GEOLOC code of the transfer port	POXCOM
POXSHP(S)	true if this is a special transfer port ship	POXCOM
PPDIST(I,J)	the J-th entry in distance table I; it contains the port-to-port distance matrix stored in triangular form; distances are extracted using subroutine DISTIM; the tables are I = 1, distances with both canals open I = 2, Panama closed, Suez open I = 3, Panama open, Suez closed I = 4, both canals closed (INTEGER*2)	DISCOM
PPROW(P)	permits computing the pointer PP for finding values of PPDIST, ARISK, or BRISK from port I to port J, with I<J: PP=PPROW(i) + j	DISCOM
PPTIME(PHASE)	the starting day of a time phase	DISCOM
PQUE(P)	TRUE if queuing is to be used at this port (LOGICAL)	PORT
PRBTYP(I)	the problem type the I-th compartment type will be placed in	STOW
PREBST(S)	indicates that ship S has been preferred at one of the levels: 0 = no preference 1 = preference for the type of ship 2 = preference for the specific ship	CYCLEA
PRED(I)	the predecessor node of node I	DCALC
PRID(I)	the port identification of the I-th port stop of a ship	PATHCOM
PR(I,PT)	the row number of the I-th entry in the solution vector of problem PT	TRANSP

Variable Name	Description	Common
PRNEW(I)	the new PR vector for one problem	REFCOM
PRSTOR(I)	temporary storage of the PR(I,PT) vector for the sailed ship	CYCLE
PRT(I)	temporary storage of the PR vector	SETUP
PSAV1-PSAV6(P)	port arrays saved prior to inserting a candidate channel into a ship's route; the arrays save arrays PCHAN, REALCH, PORDER, ACORD, PDIS, and PTIME; if the channel should not be inserted in the route, these arrays can be restored (PSAV4 is CHARACTER*1, PSAV6 is REAL)	CYCLE
PSAV7(CH)	this array saves FINISH(CH) for the candidate channel while the channel is being evaluated for insertion into a ship's route; FINISH is the time a channel completes unloading	CYCLE
PSHIP(C)	preferred ship for a cargo	CARGO2
PSHIPT(C)	preferred ship type for a cargo	CARGO2
PT8(F)	temporary arrays for reading port data	INPUT
PT	the problem type; problem type 1 is all dry cargo, problem type 2 is POL	PARETC
PTDIM	the dimension on the number of problem types	PARETC
PTEXST(PT)	TRUE if problem type PT exists (LOGICAL)	PARETC
PTIME(I)	the time a ship will be ready to leave the I-th port on its route (REAL)	CYCLE
PTNEXT(S)	the next port to be visited by ship S during simulation	SCHED
PTRIP(S)	in process trip number; used in execution planning (execution planning experiments only	SHIP
PTSEQ(S,I)	the port of the I-th assignment on ship S (INTEGER*2)	SCHED
PTSEQT(I)	temporary storage of PTSEQ(S,I)	PATHCOM

Variable Name	Description	Common
PTYPE(P,F)	type of facility of port P, facility F: 'A' = ammo 'B' = container ammo 'C' = container 'G' = general 'P' = petroleum (CHARACTER*1)	PORT
PUSH	if TRUE, cargo will be sent to its intermediate port even if it cannot continue to its port of embarkation (LOGICAL)	PARETC
PW	the peace/war indicator; 1 = peace, 2 = war	PARETC
PWHEN(S)	the day the ship started the in-process trip (execution planning experiments only)	SHIP
PWRMR(P,T,FT)	amount of fuel needed at port P, time period T, fuel type FT; the time periods are in fixed 10-day intervals	BUNK
Q1(I)	short tons of the I-th LISTOUT record	FINI
Q2(I)	measurement tons of the I-th LISTOUT record	FINI
Q3(I)	square feet of the I-th LISTOUT record	FINI
Q4(I)	number of passengers of the I-th LISTOUT record	FINI
Q5(I)	hundreds of barrels of the I-th LISTOUT record	FINI
QED(P)	ship-days of queuing at port P (REAL)	SCHED
QNEO	number of evacuees (NEO)	PARETC
QSHIP(S)	accumulated queue time of a ship at a port	SCHED
QSUM(C)	quantity balance for cargo C	POSTCM
QTOTAL(P,F,I)	total nominal shiploads at port P, facility F, during time period I (REAL)	QUEUE
RANK(S)	the rank of a ship, based on VAL(S); the best (that is large, fast) ships have lower rank	SHIP
REALCH(I)	the channel number of the I-th activity of a ship being routed	CYCLE

Variable Name	Description	Common
RECALC	indicates how much distance recalculation is necessary 0 - no recalculation required 1 - recalculation only ship availability dates (SADS) 2 - recalculate port-to-port distances and SADS	DCALC
RECDIM	dimension of the number of records on the LISTOUT file	PARETC
RECNO	the number of records in the route file	PTR
RECOST(K,L)	cost matrix for the K-th late trip, L-th free ship	REUSE
RECRD1(F,PN)	temporary storage of *.PTR files for use in building the SHIPS.PTR file...F=1: ALLOPEN, F=2: PANSHUT, F=3: SUESHUT, F=4: ALLSHUT	PTR
RECRD2(F,PN)	same as above for the last record pointer	PTRr
REFSAD(S)	the smallest ship availability date of ship S	SAD
RELEAS(C)	amount of cargo C that is released from its independent leg	SCHED
REQCON(C)	special cargo indicator: 'A' = AFOE 'C' = CONSOL ' ' = not special (CHARACTER*1)	CARGO1
RID(C)	identification of cargo C	CARGO1
RISK(LAT,LON)	The risk of attrition for each day a ship is at latitude LAT, longitude (LON). If a negative number, the point is in an ocean avoidance area.	RISKCOM
RKEEP(ROW)	TRUE if a row of the transportation problem is to be kept in the general network (ship loading) problem (LOGICAL)	SETUP
RLD(C)	cargo ready to load date at origin	CARGO2
RLDALD	maximum time between RLD and ALD; may be used to automatically advance the ALDs	PARETC
RNAME(I)	the 8-character name of the risk file applicable to the I-th change	OCEAN
ROOM(S,ST)	the space left in ship S, stow type ST	NOPT

Variable Name	Description	Common
ROUTED(PT)	total quantity of cargo routed in problem type PT	PARETC
ROWDIM	dimension on the number of rows in the transportation problem	PARETC
ROWJ(I)	the row number in the transportation problem corresponding to the I-th variable of the general network problem	SETUP
ROWNEW(ST,TR)	the row of stow type ST, trip TR of the target ship	REFCOM
ROWORD(I)	the I-th entry of this array contains the solution vector index (the first index of the X-vector) of the transportation problem sorted by rows	TBASIS
ROWOS(I)	the row of the I-th assignment on a ship	CYCLEA
ROWPT(S,ST,TR)	the row number of ship S, stow type ST, trip TR; this is a temporary pointer used in building the final solution passed to the ship loading subsystem	CYCLE
ROWST(ROW)	the index of the solution vector for the first solution in a row	TBASIS
ROWTOT(ROW)	the index of the solution vector for the last solution in a row	TBASIS
ROWV(ROW)	row numbers associated with a ship, used in NXTBST	NOPT
RPORT(P)	true if a port is a POE and should be reported	RLCOM
RSKOS(I)	accumulates the risk to the cargo; it is the probability that the cargo survives	SCHED
RUNTYP	type of run "A": pre-aggregation cargos "C": closure; regular sail run "D": distance computations only "F": find ships (locate them on route)* "L": lift capability "P": project ships forward from present position* "R": required lift computations "S": ship availability dates and distances "X": transfer port, first pass* *not required as input...hardwired in main programs	PARETC

Variable Name	Description	Common
SAD(S,P)	the ship availability date of ship S at port P (INTEGER*2)	SAD
SADCAN(S,P)	the canal used by a ship in reaching port P: 'P' = Panama Canal 'S' = Suez Canal 'B' = both ' ' = neither (CHARACTER*1)	SAD
SADGET	the file name from which ship availability dates are to be read (CHARACTER*7)	PARETC
SADMUL	multiplier for average distance for projecting ship availability dates beyond the first trip (REAL)	PARETC
SADONE(S,P)	the first ship availability date; unaltered throughout execution (INTEGER*2)	SAD
SADSAV	file name for saving ship availability dates (CHARACTER*7)	PARETC
SAILS(S,TR)	TRUE if a ship trip is to sail; FALSE otherwise (e.g., if the ship is inadequately loaded) (LOGICAL)	SCHED
SAILTM(S)	the time a ship sails after loading cargo	ARRDEP
SASHT(SHPT)	the number of ships available by ship type	REPORT
SBEAM(S)	ship beam, in feet	SHIP
SCARGO(S,TR, CMP,ST)	cargo unloaded from ship S, trip TR, compartment CMP, stow type ST	SREPT
SDIM	dimension on number of ships	PARETC
SDLAT(S)	latitude of a ship's original destination (REAL)	SHIP
SDLON(S)	longitude of a ship's original destination (REAL)	SHIP
SDRAFT(S)	draft of a ship, fully loaded	SHIP

Variable Name	Description	Common
SEED	the seed of the random number generator used in attrition	PARETC
SEQ(C)	if cargo C must be loaded on more than one ship, each loading is given a different sequence number (INTEGER*2)	SCHED
SEQE(C)	sequence number of cargo C; counted as records are read from the EXCLOUT file	POSTCM
SEQMAX	the maximum number of entries in the simulation sequence vectors (CHSEQ, AMSEQ, etc.)	SCHED
SEQOS(S,I)	cargo sequence number of the I-th cargo loaded on a ship; see SEQ(C) (INTEGER*2)	SCHED
SFCONT	a multiplier to convert ship capacity from square feet to container stowage (REAL)	PARETC
SFFRE(ST)	during the trial loading of a channel onto the free ship, this is the stow factor of that channel on the ST-th compartment	REUSE
SFST(ST)	temporary stow factor for the ST-th compartment (REAL)	NOPT
SHEATH(ROW)	TRUE if the compartment represented by the row has been sheathed; used in routing (LOGICAL)	CYCLE
SHFAV(S)	favorability of ship S	REFCOM
SHPCAP(SHPT, ST)	for ship type SHPT, stowage type ST contains: -1 if the ship does not have that type of stowage 0 if it has the stowage, but it is incidental 1 if this is a major stowage area for the ship	SHIP
SHPCLS(CLS, SHPT)	TRUE if a cargo class can be carried on a ship of type SHPT (LOGICAL)	STOW
SHPTNR(SHPT)	the ship type identifying number of the SHPT-th ship type; read from the SHPTDAT file; see Table 5.4	SHIP
SHPUSE(S,PT)	TRUE if ship S is to be used in the problem type PT (LOGICAL)	SHIP

Variable Name	Description	Common
SHTDIM	ship type dimension	PARETC
SID(S)	identification of ship S	SHIP
SIMSHE(S,STX)	TRUE if ship S, stow type ST (ST=1,3), is sheathed for ammunition loading (LOGICAL)	SCHED
SLACK(S,TR,ST)	the amount of slack capacity of ship S, trip TR, stow type ST	SCHED
SLENG(S)	ship length in feet	SHIP
SLIP(S,TR)	amount of schedule slippage allowed for ship S, trip TR	SCHED
SLIPUS(S,TR)	amount of slip used on ship S, trip TR	SCHED
SLIST(ROW)	the ship compartment that is a candidate for carrying the cluster	NOPT
SMAT(S,PT)	the row number of the first compartment of ship S in the transportation matrix	TRANSP
SNARIO	scenario description (CHARACTER*12)	PARETC
SOL(I)	the I-th column of those assigned to a ship compartment	CYCLE
SOLGET	the file name of a saved routing solution (CHARACTER*7)	PARETC
SOLSAV	the file name for saving the routing solution (CHARACTER*7)	PARETC
SOPLAN(S,I)	true if ship S can be used in OPLAN I	MISCOM
SOPLAN(S,PL)	true if chip S can be used in plan PL	MISCOM
SPAN(I)	the length of a link in nautical miles	DCALC
SPEED(S)	ship speed in nautical miles per day (REAL)	SHIP
SPLAT(S)	initial ship location latitude (REAL)	SHIP

Variable Name	Description	Common
SPLON(S)	initial ship location longitude (REAL)	SHIP
SPOINTS(S,I)	pointers to the ship route lat/lons for ship S; 1=1: 1st index; I=2: 2nd index	PTR
SPREP(S)	miscellaneous ship preparation delays, including tanker cleaning delay, in days	SHIP
SQFTMT	the ratio of dry cargo quantities expressed in measurement tons to those same cargo quantities expressed in square feet; this ratio is used for capacity conversions (REAL)	PARETC
SRANGE(S)	ship's fuel range in nautical miles	SHIP
STAT(S)	the code for the next activity on ship S 'S' = sail 'D' = dock 'L' = load 'U' = unload 'N' = null (no activity) 'B' = bunker 'I' = intermediate unload 'F' = finished (CHARACTER*1)	SCHED
STATUS(I)	node status in the shortest path algorithm 0 = unexamined 1 = tentative length found -1 = shortest path found	DCALC
STDIM	dimension on the number of storage types	PARETC
STEXST(ST)	TRUE indicates the existence of cargo of stowage type ST in this problem type (LOGICAL)	PARETC
STFRAC(S)	fraction of capacity used on ship S for all trips (REAL)	REFCOM
STONS(C)	short tons of a cargo (if applicable)	CARGO2
STOW(CLS,SHPT, ST)	stow factor for cargo class CLS, ship type SHPT, stow type ST (REAL)	STOW

Variable Name	Description	Common
STOWCG(CLS)	approximated stow factor for cargo class CLS (read from the last row in the stow factor table in Appendix E) used to adjust the demands in the transportation problem (REAL)	STOW
STOWED(CLS, SHPT,ST)	measurement tons of cargo class CLS in stowage ST on ships of type SHPT	SCHED
STOWST(SHPT,ST)	approximated stow factors by compartment (read from the last column in the stow factor table in Appendix E) (REAL)	STOW
STRAND(ST)	the amount of dependent cargo of stowage type ST stranded at the intermediate port	SCHED
STSEQ(S,I)	ship stowage type for the I-th channel assignment for ship S (INTEGER*2)	SCHED
STSEQT(I)	temporary storage of STSEQ	PATHCOM
STSHT(SHPT)	the number of ship trips sailed by ships of type SHPT	REPORT
STSKED(S,TR,I)	ship stow type used for the I-th channel assigned to ship S, trip TR (INTEGER*2)	SCHED
STTOT	total measurement tons by stow type	SREPT
STUFF(CLS, CELL)	histogram cells by cargo class; the amount of cargo delivered accumulated by days deviation from on-time	POSTCM
STUFIN(S,ST)	stuffing factor...used in lift capability analysis; gives the average load expected in each compartment	SHIP
STUSE(I)	the stow types of the compartments of the target ship	REFCOM
STV(R)	the stow type of the R-th row of a ship	NOPT
STYPE(S)	type of ship S	SHIP
SUEPAS(S)	true if a ship has gone through the Suez canal on the way to its present location	LOCATE
SUEPTR(P,K)	first and last records for the route from port P to Suez	PTR

Variable Name	Description	Common
SUEX(S)	true if ship S used (or expects to use) the Suez Canal	LOCATE
SUEZB	maximum beam for passage through the Suez Canal	PARETC
SUEZD	maximum draft for passage through the Suez Canal	PARETC
SUEZL	maximum length for passage through the Suez Canal	PARETC
SUPPLY(ROW,PT)	total supply in a row in problem type PT	TRANSP
SUSHT(SHPT)	the number of ships of type SHPT used	REPORT
SWAB	TRUE to use optimization version A; FALSE for version B (LOGICAL)	PARETC
SWGNC	TRUE if the general network code is to be used for ship loading (LOGICAL)	PARETC
SWNOPT	TRUE if the heuristic routing is to be used (LOGICAL)	PARETC
SWSEL	TRUE if the heuristic routing is used to select ships for use in the optimization routing (LOGICAL)	PARETC
TOMAX(CH)	the maximum ALD for including a cargo in channel CH	CHVAR
TOMIN(CH)	the minimum ALD for including a cargo in channel CH	CHVAR
T1(I)	the beginning of time interval I	CHANCOM
T2(I)	the end of time interval I	CHANCOM
T8(SHPT)	MT carried by ship type SHPT (REAL)	REPORT
T9(SHPT,CL)	100 times MT of cargo class CL carried by ship type SHPT (REAL)	REPORT
TARR(I)	arrival time at port, not including docking or queuing time, for the I-th activity read from LISTOUT (REAL)	ARRDEP

Variable Name	Description	Common
TARRIV(S)	arrival time at a port for ship S (REAL)	SCHED
TCAP(P,F,J)	day on which the j-th change in port throughput occurs at port P, facility F	PORT
TCAPT(P,F,I)	the beginning time of the I-th interval (used in conjunction with MCAPT to evaluate port capacity availability) at port P, facility F (REAL)	SCHED
TCLINF(CL)	MT of infeasible requirements by cargo class	REPORT
TCLSEA(CL)	ton-days of earliness by cargo class	REPORT
TCLSL(CL)	MT scheduled late by cargo class	REPORT
TCLSOT(CL)	MT scheduled on time by cargo class	REPORT
TCLSTD(CL)	ton-days of lateness by cargo class	REPORT
TCPRT(P,F,I)	MT of cargo through port P, facility F I = 1 for POE I = 2 for POD	REPORT
TDIM	the maximum days in schedule dimension	PARETC
TDKEEP	the current value of the ton-days of lateness criterion	CYCLE
TDSHT(SHPT)	total distance sailed by ship type	REPORT
TEMP(CC)	temporary arrays for reading load and unload rates	INPUT
TERM(I)	the terminus node of a link in the shortest path algorithm	DCALC
TESU(ST)	total estimated ST stowage used	CYCLE
TGTFRC(CH)	target fraction of channel CH to be loaded (real)	LIFCAP
TIMCAP(P,SHPT, T)	time-dependent capacity of ships of type SHPT needed at port P in time period T (required lift calculations only)	RLCOM
TIME1(I)	the time the I-th load can begin (avoid docked ship delays for cargo ALDs) (REAL)	SCHED

Variable Name	Description	Common
TIME2(I)	approximate time the I-th load is completed (REAL)	SCHED
TINDEX	index for building the minisolution vectors for a single ship sailing	NOPT
TINDS(ROW)	a temporary value of INDS(ROW,PT)	CYCLE
TINDST(ROW)	a temporary value of INDST(ROW,PT)	CYCLE
TINDTR(ROW)	a temporary value of INDTR(ROW,PT)	CYCLE
TLAT(I)	latitudes along an arc	ARCCOM
TLBOAT(P,SHPT)	number of nominal ships of type SHPT needed at port P	RLCOM
TLCAP(P,SHPT)	capacity of ships of type SHPT needed at port P (required lift calculations only)	RLCOM
TLON(I)	longitudes along an arc	ARCCOM
TNEXT(S)	the time of the next event for ship S (REAL)	SCHED
TONDAY(S,TR)	ton-days of lateness on shipS, trip TR	REUSES
TONDER(S)	ton-days of earliness on ship S	REFCOM
TONDLA(S)	ton-days of lateness on ship S	REFCOM
TONS(S,TR)	tons of cargo carried on ship S, trip TR	REUSES
TOPOD	completion time of the last unload of a trip	CYCLE
TOT(CL)	total amount of cargo of class CL	POSTCM
TOTED(CL,SHPT, ST)	cargo amount--MT, PAX, or CBBLS--of class CL carried by ships of type SHPT in compartment stowage type ST	POSTCM
TOTITS	the total number of iterations to solution	VECTOR
TRCLUS(I)	the trip carrying the I-th cluster	REFCOM
TRDIM	the maximum number of ship trip's dimension	PARETC
TRIP(S)	the trip number being considered for ship S	SHIP

Variable Name	Description	Common
TRIPL(I)	the trip number of the I-th port	REFCOM
TRRSEQ(S,I)	the reported trip number (INTEGER*2)	SCHED
TRRSQT(I)	temporary storage for TRRSEQ	PATHCOM
TRSEQT(I)	temporary storage for TRSEQ	PATHCOM
TRSEQ(S,I)	solution vector trip number for the I-th assignment on ship S (INTEGER*2)	SCHED
TRSTOR(I)the	ship trip number of the I-th sailed cargo	CYCLE
TSPRT(P,F,I)	tons scheduled through port P, facility F I = 1 for POE I = 2 for POD	REPORT
TTCLS(CL)	total tons by cargo class	REPORT
UNIT(C)	unit identification of cargo C (CHARACTER*6)	CARGO2
UNLOAD	nominal unload time (REAL)	PARETC
UNPORT	TRUE if port length, width, and depth are unconstrained (LOGICAL)	PARETC
UNRATE(CL, SHPT,ST)	cargo class CL unload rate when on ship type SHPT, stowage type ST	STOW
USE1(S)	date of first use of ship S	RLCOM
VAL(S)	the 'value' of a ship (speed*capacity)	SHIP
VALUE(I)	vector of values (functions of distance) from the source to very node I (calculated in subroutine SPA)	DCALC
WAIT	the number of days a ship will wait for the arrival of the first leg of the intermediate cargo	PARETC
WCOST(I)	the cost of the I-th arc (REAL)	WARCS

Variable Name	Description	Common
WFLOW(I)	the flow on the I-th arc (REAL)	WARCS
WHEAD(I)	the head node of the I-th arc	WARCS
WMULT(I)	the multiplier of the I-th arc (REAL)	WARCS
WTAIL(I)	the tail node of the I-th arc	WARCS
X(I,PT)	the cargo amount of the I-th solution in the solution vector	TRANSP
XDIM	the maximum length of the solution vector dimension	PARETC
XINUM(PT)	the number of elements in the solution vector	TRANSP
XNEW(I)	the X-vector after REFINE	REFCOM
XOS(I)	the amount of the I-th channel on a ship	CYCLEA
XSTORE(I)	temporary storage of X(I,PT) for a sailed ship	CYCLE
XT(I)	temporary X-vector	SETUP
XTEMP(CC)	temporary arrays for reading stow factors	INPUT

APPENDIX G

MESSAGES

This appendix will display the messages that might be encountered while running the SAIL model. Although some messages may appear on several files, they may only be described once herein. The Exception Report (file 13, EXEPOUT) will command major emphasis, because information on this report is an integral part of interpreting SAIL's results. Problems causing run termination will usually be seen on both the on-line and the Exception Report. Errors likely to cause unexpected or incomplete results will also be seen in both files.

Two sections will be presented: first, fatal errors (i.e., errors that terminate a SAIL run); and second, messages that may require programming attention, may cause the planner to review his results, or may simply relay information about the progress of the run.

In general, messages will identify where they are written. For example, "MAIN:INCGO: POE ... " would be a message generated from coding found in block MAIN, subroutine INCGO.

FATAL ERRORS

When an incorrect or unexpected combination of data files has been compiled, identification fields will mismatch and one of the following three errors will occur.

MAIN:INCGO: OPLAN, CYCLE, RUN, MODULE ON A REQUIREMENT RECORD
DOES NOT MATCH THE PARAMETER RECORD;
RUN ABORTED

MAIN:INSHIP: OPLAN, CYCLE, RUN, MODULE ON A SHIP RECORD
DOES NOT MATCH THE PARAMETER RECORD;
RUN ABORTED

MAIN:INPORT: OPLAN, CYCLE, RUN, MODULE ON A PORT RECORD
DOES NOT MATCH THE PARAMETER RECORD;
RUN ABORTED

The following errors occur when the model dimensions are inadequate to process the current problem.

MAIN:INCGO: CARGO DIMENSION EXCEEDED; MAXIMUM IS nnnnn

MAIN:INSHIP: A STOW TYPE IN THE SHPTDAT FILE IS
GREATER THAN STDIM: nnn

MAIN:INSHIP: TOO MANY SHIPS--DIMENSION IS nnn

MAIN:INPORT: TOO MANY PORTS ON THE INPUT FILE
THE DIMENSION IS: nnn

MAIN:SPLIT: SPLITTING CARGOS FOR INTERMEDIATE STOPS
HAS CAUSED THE NUMBER OF CARGOS TO EXCEED
THE CARGO DIMENSION:nnnnn

CHANEL: MAXIMUM NUMBER OF CHANNELS (nnnn) EXCEEDED

MATGEN: NUMBER OF ROWS EXCEEDS DIMENSION, PROBLEM:nn

MATGEN: TOO MANY ROWS (nnnn) AND COLUMNS (nnnn) FOR
THE DIMENSION OF THE SOLUTION VECTOR (nnnnn)

DIST:INLINK: NUMBER OF LINKS EXCEEDS DIMENSION nnnn

DIST:ADPORT: ADDING NODES FOR PORTS
EXCEEDS NODDIM nnnnn

DIST:ADPORT: ADDING LINKS FOR PORTS
EXCEEDS LNKDIM nnnnn

DIST:INNODE: NUMBER OF NODES (nnnn)
EXCEEDS NODDIM (nnnn)

DIST:DISFIL: DISTANCE VECTOR TOO LONG;
NUM=nnnnn DISDIM=nnnnn

MATGEN: NR OF COLUMNS EXCEEDS DIMENSION, PROBLEM:nn

ACYCLE:ASAILM: SOLUTION DIMENSION iii EXCEEDED
SCHEDULING TERMINATED

NCYCLE:NSAILM: SOLUTION DIMENSION nnn EXCEEDED
SCHEDULING TERMINATED

NOPT:STASH: OVERFLOW DIMENSION XDIM, PT=n

LP84:GNCDAT: NUMBER OF ARCS EXCEEDS DIMENSION

LP84:GNCDAT: NUMBER OF NODES EXCEEDS DIMENSION

LP84:GNCOUT: OVERFLOWING XDIM, PT=n

PATH:SKEDIN: SHIP nnnn HAS MORE THAN nnn STOPS
ON TRIP nn; INCREASE CHSDIM

PATH:SKEDIN: NUMBER OF EVENTS TOO LARGE;
INCREASE EVDIM

PATH:PATH: NUMBER OF EVENTS IN SEQUENCE ARRAYS
EXCEEDS EVDIM; SHIP nnnn, TRIP nnn

SIMULA:PORTTM: FATAL QUEUE TABLE ERROR

Note: The following message will not be fatal on simulation pass zero, the pass made to get timing for the ship loading subsystem.

SIMULA:LOADUP: NR OF CARGOS ON SHIP EXCEEDS DIMENSION

SIMULA:ASSURE: TOO MANY CHANNELS ADDED TO CHSKED
FOR SHIP,nnnnn TRIP,nnnnn;
INCREASE CHSDIM

REFINE:REFIN1: IN BUILDING A NEW X VECTOR,
THE INDEX HAS GOTTEN LARGER

THAN XDIM; ABORT

REFINE:REFIN2: IN BUILDING A NEW X VECTOR,
THE INDEX HAS GOTTEN LARGER
THAN XDIM; ABORT

Proper maintenance of the SAIL model will undoubtedly require complete understanding of the Semi-Permanent Data Files (see Paragraph 5.2.2.3). Files LOADDAT, UNLDDAT, STOWDAT, and SHPTDAT describe SAIL ships and their relations with port facilities. Inconsistencies in this information will produce the following errors.

MAIN:INSHIP: TOO MANY SHIP TYPES ON FILE SHPTDAT

MAIN:INSHIP: SHIP TYPE IN RECORD nnn OF SHIPDAT FILE DOES NOT
MATCH ANY SHIP TYPE OF SHPTDAT FILE

MAIN:INSTOW: BAD SHIP TYPE IN LOADDAT FILE: nn

MAIN:INSTOW: BAD SHIP TYPE IN UNLDDAT FILE: nn

MAIN:INSTOW: BAD SHIP TYPE IN STOWDAT FILE: nn

The remaining messages are traps built into the code to recognize potential problems resulting from two sources: (1) entering untested code segments or (2) incorporating new code into SAIL. Fully tested code should not produce these errors. A brief comment will be given to give direction to debugging effort in the event these messages are encountered.

In general, POL can be carried only on tankers and "dry cargo" cannot be shipped on tankers. Because shipping resources are not shared, an opportunity exists for reducing the size of the resource allocation problem. It should be more efficient to break the problem into two or more subproblems, solve each smaller resource allocation problem, merge the results for simulating the port interactions, and report the plan as a whole. Currently SAIL creates two subproblems (referred to as problem types), tanker only and dry cargo only. A "problem type" subscript is used to separate them, and if that subscript is tainted the following messages could occur.

MATGEN: WRONG PROBLEM TYPE FOUND: nnn

MATGEN:MATREV: INCORRECT PROBLEM TYPE nn
FOR INTERNAL SHIP NR nnnn

The following message will appear if CHANEL discovers invalid port numbers.

CHANEL:CHANEL: BAD PORT NUMBER FOUND FOR CARGO: nnnnn
POE: nnnn POI: nnnn POD: nnnn

Subscripting problems while accessing interport distances (bad port numbers) will generate the

next two messages.

DIST:DIST: FIRST PORT:nnnn IS OUT OF RANGE(1-nnnn)

DIST:DIST: SECOND PORT:nnnn IS OUT OF RANGE(1-nnnn)

There are four distance tables (one for each combination of canal settings) containing the interport distances.

DIST:DIST: BAD TABLE OR TABLE ENTRY FOUND;
TABLE (TAB):nnn ENTRY (N):nnnnn

The ocean grid consists of ocean points, or nodes, connected by links defining all possible paths ships could be on once connected to the grid. The next message points to inconsistent data in the NODEDAT and LINKDAT files; in this case, a link is defined as connecting two nodes, at least one of which is not defined in the NODEDAT.

DIST:INLINK: NODE ID FOR LINK aaaa NOT FOUND IN NODEDAT

A ship or port latching onto the grid requires finding the nearest node and the distance to that node. Those distances are calculated as great circle distances.

DIST:GCD: GREAT CIRCLE DISTANCE ERROR nnnnn

DIST:NEAR: ERROR IN FINDING CLOSEST NODE
LATITUDE: n.nnnnnnn
LONGITUDE:n.nnnnnnn

Selecting the interport shortest paths is accomplished in subroutine SPA where the terms "pivot location" and "forward star indexes" are used. Knowledge of the Shortest Path Algorithm is required to troubleshoot the following messages.

DIST:SPA: PIVOT LOCATION NOT FOUND

DIST:SPA: INDEXES IN FORWARD STAR ARE WRONG FOR SOURCE:nnnnn

A breakdown in the prime cargo selection process has occurred if the next message is seen. It means that there is still cargo assigned to unsailed ships, but criteria for prime selection cannot be met.

ACYCLE:APRIME: FAILURE TO FIND A BEST SHIP;
SHPBST= nn

NCYCLE:NPRIME: FAILURE TO FIND A BEST SHIP;
SHPBST= nn

The nature of the transportation model requires exactly N elements in the solution vector, where N is one less than the sum of the number of supplies and the number of demands. In the unlikely event that this need is violated, the following message would occur.

SOLVE:FIRSTX: BASIC SOLUTION HAS INCORRECT NUMBER
OF ENTRIES; DIFF= nnnnn

This is a subscripting problem. Valid port numbers are determined by input data. Valid facilities, or port types, and their subscripts are: 'A'= 1, 'B'= 2, 'C'= 3, 'G'= 4, 'P'= 5.

SIMULA: INVALID PORT OR FACILITY TYPE FOUND FOR
A SHIP TRYING TO DOCK; SEE TESTOUT FOR DETAILS

Valid status codes are: 'F', 'L', 'U', 'I', 'D', 'S', 'B', 'N'.

This error is unlikely to occur, but if it does, programming assistance will be required to resolve it.

SIMULA: STATUS IN ERROR FOR INTERNAL SHIP NR: nnn

Valid activity codes are: 'L', 'U', 'I', 'B'.

SIMULA:STATUS: SHIP nnnn SEQUENCE INDEX nnn
HAS A BAD ACTIVITY CODE: a

SIMULA:PORTTM: TIME RAN OUT WITHOUT ENOUGH THROUGHPUT

The following message will appear if the number of cargo types is changed without changing subroutine CHORDR. The vector GETGRP must be redimensioned and the group definition must be supplied in the DATA statement. Group definitions are given in comments at the beginning of the subroutine.

CHANEL:CHORDR: DIMENSION OF CARGO TYPES (CTDIM) IS
INCONSISTENT WITH GROUP DEFINITIONS
IN THIS SUBROUTINE; RECODING IS
REQUIRED

NONFATAL MESSAGES

The first nonfatal message type is that dealing with "bad data," that is, missing key data elements or inconsistent data. These messages indicate the offending record, that which is either in error or has no supporting data. A requirement record could indicate a POE that is not recognized (NOT FOUND). The problem could be that the port identification was misrepresented on the record or the port definition on PORTDAT could be missing or incorrect. The following is a list of error messages of this type.

MAIN:INCGO: CARGO TYPE (aa) FOR RECORD nnnn IS INCORRECT
REQUIREMENT IS EXCLUDED FROM PROBLEM

MAIN:INCGO: FACILITY TYPE a FOR RECORD nnnn IS INCORRECT
GENERAL PORT ASSUMED

MAIN:INCGO: PREFERRED SHIP NOT IN SHIP FILE: nn
REQUIREMENT RECORD: nnnnn; DEFAULT TO NONE

MAIN:INCGO: PREFERRED SHIP TYPE NOT IN SHIP FILE: nn
REQUIREMENT RECORD: nnnnn; DEFAULT TO NONE

MAIN:INCGO: POD aaaa NOT FOUND FOR REQUIREMENT nnnnn
REQUIREMENT IS DELETED

MAIN:INCGO: POE aaaa NOT FOUND FOR REQUIREMENT nnnnn
REQUIREMENT IS DELETED

MAIN:INCGO: POI aaaa NOT FOUND FOR REQUIREMENT nnnnn
NO INTERMEDIATE STOP IS MADE

For cargo codes '1V', '2N', '2F', or '30', if only measurement tons or square feet is specified, the other will be estimated. The following message appears when this occurs.

MAIN:INCGO: MISSING QUANTITY DATA FILLED IN
FOR CARGO TYPE: aa RECORD: nnnnn

The second message type deals with inconsistencies in the data, such as passenger movement requirements with no passenger ships in the ship file (SHIPDAT). Some of these could point to inconsistencies in internal tables, such as a port facility found in table FACIDAT could have no loading rates in the LOADDAT file.

MAIN:INSTOW: FACIDAT TABLE HAS ERROR IN CODE TYPE: FACILITY DELETED

MAIN:SPLIT: INTERMEDIATE STOP CARGO NUMBER nnnnn

IS UNAVOIDABLY nnn DAYS LATE

MAIN:INFEAS: FACIDAT AND LOADDAT TABLES DISAGREE

CHANEL:ELIM: THERE ARE NO SHIPS THAT CARRY CARGO nnnn

CHANEL:ELIM: CARGO nnnn WILL BE TOO LATE
EARLIEST POSSIBLE DELIVERY: nnnn
LAD: nnn
SPLIT CARGO NUMBER nnnn IS UNAVOIDABLY nnn DAYS LATE

CYCLE:QWAIT: IMPROPER FACILITY FOUND IN A QUEUE
LOOKUP; SEE TESTOUT FILE FOR DETAILS

CYCLE:QWAIT: IMPROPER PORT/FACILITY FOUND IN A QUEUE
LOOKUP; SEE TESTOUT FILE FOR DETAILS

DIST:STAR: NODE nnnn HAS NO LINKS

The third message type deals with the problem dimensions conflicting with the model dimensions. Some of these can be handled in SAIL and the run will continue, but others are fatal errors. The following are not fatal errors.

CHANEL:INTERV: REQUIRED TIME INTERVALS EXCEED DIMENSION
LAST INTERVAL EXTENDED AS REQUIRED
INTERVAL DIMENSION (INTDIM) :nnnnnn
TIME DIMENSION (TDIM) : nnnnnn
EARLY INTERVAL WIDTH (INT1) :nnnnnn
DIVISION DATE (DIVDAT) : nnnnnn
LATER INTERVAL WIDTH (INT2) :nnnnnn

DIST:INLINE:
NOTE THAT LINE FILE CONTAINS AT LEAST AS MANY
RECORDS AS IT IS DIMENSIONED FOR...SOME LINES
MAY HAVE BEEN LOST...CHECK LINE FILE FOR SIZE

CYCLE:ENRICH: OVERFLOWING XDIM, PT=nn
ENRICHING TERMINATED

CYCLE:CHEVAL: WARNING! TOO MANY CANDIDATE
CHANNELS; INCREASE CHSDIM

SIMULA:INSERT: KDIM EXCEEDED

BUNKER: OVERFLOW OF SHIP EVENTS WHILE BUNKERING
BUNKER ACTIVITY NOT INSERTED
SHIP IS nnnnn

REFINE:REFIN1: SHIP nnnn CANNOT BE REFINED DUE
TO EVDIM/2 BEING TOO SMALL

REFINE:REFIN2: SHIP nnnn CANNOT BE REFINED DUE
TO EVDIM/2 BEING TOO SMALL

REFINE:DUMSHP: XDIM EXCEEDED; FURTHER DUMMY SHIP
ASSIGNMENTS SUSPENDED

Another message type echoes parameter settings and other run conditions selected by the planner or built from the data.

MAIN:INPARA: SOME PARAMETER SETTINGS:
PORT SIZE RESTRICTIONS IGNORED? T/F
MAXIMUM DAYS WAIT AT INTERMEDIATE PORTS
BEFORE STRANDING CARGO: nnnn
INTERMEDIATE STOPS OF LESS THAN nnn DAYS
WILL NOT UNLOAD CARGO
CARGO TO BE SENT TO INTERMEDIATE PORTS EVEN IF
IT CANNOT BE DELIVERED TO THE POD? T/F
MAIN:INCGO: OVERALL SQUARE FOOT TO M-TON CONVERSION FACTOR:
n.nnnnnnnn

THE FOLLOWING SHIPS WILL NOT GO THROUGH THE CANALS INDICATED:
nnnnn PANAMA
nnnnn SUEZ
.
.
.

The next class of messages report problems, or potential problems, encountered during the SAIL run. These are usually informative and require no maintenance action.

CHANEL:PODQ:
MAXIMUM THROUGHPUT AT PORTnnnnn IS EXCEEDED IN INTERVALnnnnn
REQUIREMENTS ARE nnnnnn.nn, AVAILABLE IS nnnnnnnnn

CHANEL:POEQ:
MAXIMUM THROUGHPUT AT PORT nnnn IS EXCEEDED IN INTERVAL nnnn
REQUIREMENTS ARE nnnnnn.nn, AVAILABLE IS nnnnnnnnn

SIMULA: THE FOLLOWING SHIPS (IF ANY) WERE ELIMINATED DUE TO LIGHT LOADS

SID	TR	AMOUNT	FRACTION
nnnnn	nn	nnnnnn.	nn.nn
nnnnn	nn	nnnnnn.	nn.nn
	.		

.
.

SIMULA: SHIP nnnnn IS LEAVING nnnnnnn UNITS OF
CARGO STRANDED AT PORT aaaa

SIMULA: SHIP MUST WAIT DUE TO PORT AVAILABILITY DATE
SHIP-ID:nnnnn AT PORT: aaaa GOING TO: aaaa
PAD:nnn PAD DELAY:nnn

SIMULA: AFOE SHIP OR CONSOL SHIP MUST WAIT DUE TO CARGO EAD
SHIP-ID:nnnnn AT PORT: aaaa GOING TO: aaaa
EAD:nnn EAD DELAY:nnn

BUNKER: DIVERSION FAILED
SHIP: nnnnn
TRIP: nn
LAST BUNKER PORT MADE: aaaa

SIMULA:CHKCAP: TNOWT EXCEEDS MAX PORT AVAILABLE TIME

Next, a list of messages about problems found that require resolution.

When the transportation algorithm exceeds the iteration limit, the suboptimal solution at that point is returned. If this message is infrequent, no action is needed, but if it were to occur repeatedly during a run, the limit could be raised.

SOLVE: ITERATION LIMIT EXCEEDED nnnnn
SOLUTION PROCEDURE CONTINUES

Problems estimating probabilities in the queuing parameters might produce either of the next two messages.

CHANEL:PODQ: PZERO INFEASIBLE: nnnn.nnnnn
SET TO ZERO

CHANEL:PODQ: PROBABILITY OF BUSY PERIOD= nnn
SET TO ONE

PARX will be displayed when PRIME must alter costs to drive cargo from future trips to current trips.

CYCLE: PARX= nnnn

At the conclusion of a SAIL run, there should be no cargo left on board ships (exception: ships marked 'dead in the water' due to bunkering failure may load cargo that is not unloaded).

```
SIMULA: CARGOS LEFT ON SHIP   nnn
      INTERNAL SHIP NUMBER:  nnn
      SHIP TYPE (STYPE):   nn
      NR OF CARGOS ON SHIP (NUMONS):nnnnn
  CARGO AMOUNT   STOW    X    EXTRA  CLASS    POD
            USED  INDEX            NUMBER
  nnnnn  nnnnn  nnnnn  nnnnn  nnnnn  nnnnn  nnnnn
  nnnnn  nnnnn  nnnnn  nnnnn  nnnnn  nnnnn  nnnnn
                  .
                  .
                  .
```

The LOADDAT and UNLDDAT files must be kept current and consistent.

```
SIMULA:PORTTM: A LOAD RATE OF ZERO WAS FOUND;
          SEE TESTOUT FOR DETAILS

SIMULA:PORTTM: AN UNLOAD RATE OF ZERO WAS FOUND;
          SEE TESTOUT FOR DETAILS
```

Should SAIL attempt to unload more cargo than is assigned to a ship, the following message will appear.

```
SIMULA:DEBARK: X GOING NEGATIVE; SEE TESTOUT
```

A final check is made in POST to ensure that all cargo has been processed, that is, loaded and unloaded, or excluded.

```
POST:POST: QUANTITY IMBALANCE FOR REQUIREMENT ID nnnnnn
        QUANTITY BALANCE (QSUM): nnnnnnn
        C=       nnnnn
        CPOINT(C)=nnnnn
        CPOI(C)=  nnnnn
```

And finally, some messages are meant to convey run results in a form that may give the planner additional insight as to the quality of the schedule or the impact of parameter selection.

```
ATTRITION: SHIP nnnn

CARGO CLASS STOWAGE MAP:
CARGO  SHIP  COMPARTMENT AMT LOAD  AMT UNLOAD
CLASS  TYPE    TYPE     M-TONS     M-TONS
  nn    nn       n     nnnnnn     nnnnnn
  nn    nn       n     nnnnnn     nnnnnn
                .
                .
                .
```

QUEUE DAYS AT PORTS:

P	QUEUE DAYS
aaaa	nn.n
aaaa	nn.n
.	
.	
.	

OTHER ON-LINE MESSAGES

SAIL has the ability to save intermediate results so that repeated parametric runs can avoid recalculation of unchanging values, such as interport distances when successive runs use the same ships and ports. SAIL will protect itself from using incorrect tables when it can.

MAIN: SOLSAV: **aaaaaaa** SOLGET: **aaaaaaa**
MAIN: WARNING, BOTH SOLSAV AND SOLGET WERE NON-BLANK
DEFAULT TO SOLGET...SOLSAV BLANKED

DIST:DISGET: WRONG DISGET FILE SPECIFIED;
DISTANCES ARE RECALCULATED

DIST:GETDIS: INCORRECT DIMENSIONS IN SADS FILE;
SADS ARE RECALCULATED

Valid codes are: 'A', 'B', 'C', 'G', and 'P'.

MAIN:INSTOW: FACIDAT TABLE HAS ERROR IN CODE TYPE: FACILITY DELETED

CYCLE:QWAIT: IMPROPER FACILITY FOUND IN A QUEUE
LOOKUP; SEE TESTOUT FILE FOR DETAILS

CYCLE:QWAIT: IMPROPER PORT/FACILITY FOUND IN A QUEUE
LOOKUP; SEE TESTOUT FILE FOR DETAILS

MAIN:SPLIT: NUMBER OF CARGOS AFTER SPLIT: nnnn

DIST:STAR: NODE aaaa HAS NO LINKS

DIST:INLINE:
NOTE THAT LINE FILE CONTAINS AT LEAST AS MANY
RECORDS AS IT IS DIMENSIONED FOR...SOME LINES
MAY HAVE BEEN LOST...CHECK LINE FILE FOR SIZE

** WATCH OUT .. THE LP SOLUTION VECTOR IS
LONGER THAN ND+NS-1

APPENDIX H

SEASTRAT SHIP AVAILABILITY COMPUTATIONS

CASE 1 - PEACETIME COMPUTATIONS FOR ACTIVE SHIP EN ROUTE

If PEACE is true and activation/preparation time = 999:

Administrative delay days
- Early warning days
+ Travel days from present position to destination position
+ Unloading days at destination Position
+ Travel days from destination position to POE
+ Tank cleaning delay days at POE (for tankers only)

CASE 2 - PEACETIME COMPUTATIONS FOR ACTIVE SHIP IN PORT

If PEACE is true and activation/preparation time = 0:
(present position = destination position)

Administrative delay days
- Early warning days
+ Unloading days at present position
+ Travel days from present position to POE
+ Tank cleaning delay days at POE (for tankers only)

CASE 3 - PEACETIME COMPUTATIONS FOR INACTIVE SHIP

If PEACE is true and activation/preparation time is not 0 or 999:

Administrative delay days
- Early warning days
+ Activation/preparation days at present position
+ Travel days from present position to POE

CASE 4 - WARTIME COMPUTATIONS FOR ACTIVE SHIP EN ROUTE

If PEACE is false and activation/preparation time = 999:

Administrative delay days
- Early warning days

+ Travel days from present position to POE
+ Unloading days at POE
+ Tank cleaning delay days at POE (for tankers only)

CASE 5 - WARTIME COMPUTATIONS FOR ACTIVE SHIP IN PORT

If PEACE is false and activation/preparation time = 0:
(present position = destination position)

Administrative delay days
- Early warning days
+ Unloading days at present position
+ Travel days from present position to POE
+ Tank cleaning delay days at POE (for tankers only)

CASE 6 - WARTIME COMPUTATIONS FOR INACTIVE SHIP

If PEACE is false and activation/preparation time is not 0 or 999:

Administrative delay days
- Early warning days
+ Activation/preparation days at present position
+ Travel days from present position to POE

PEACE-is-true if PEACE = P and fleet code is not 10 (MSC Force); otherwise, PEACE is false.

APPENDIX I

COMMONS USED IN CODE BLOCKS

The table in this appendix shows the coding blocks across the top and the common blocks down the left side. A ▲ at the intersection of a code block and a common means that the common is included in at least one routine of that code block.

CODING ▸ MODULE / COMMON ▾	MAIN	INPUT1	INPUT2	CHANEL	DIST1	DIST2	ATTRIT	MATGEN	SOLVE	CYLLE	NOPT	OPTA	LP84	GENNET	PATH	S1	BARK	S2	BUNKER	REFINE	POST	UTIL	STOP
ARCCOM					▲	▲																	
ARRDEP													▲								▲		
BUNK	▲															▲	▲	▲	▲	▲	▲	▲	
CARGO1	▲	▲		▲											▲	▲	▲	▲			▲	▲	
CARGO2	▲			▲																	▲		
CHANCOM	▲			▲				▲		▲	▲	▲	▲		▲	▲	▲	▲	▲	▲		▲	
CHOKCOM								▲		▲	▲	▲											
CHVAR				▲																			
COSTCOM								▲	▲	▲	▲	▲											
CYCLE								▲		▲	▲	▲											
CYCLEA												▲											
DCALC					▲	▲																	
DISCOM	▲			▲	▲	▲	▲															▲	
FACDIM																					▲		
FIRST									▲														
GBASIS														▲									
GCOM													▲	▲									
GCORE														▲									
GSUP													▲	▲									
INPUT		▲	▲																				
KELLY						▲																	
LCCOM																					▲		
LOCATE						▲																	
MISCOM	▲		▲																			▲	
NOPT								▲			▲	▲											
OCEAN					▲	▲	▲																
PARETC	▲	▲	▲	▲	▲	▲	▲	▲		▲	▲	▲	▲		▲	▲	▲	▲	▲	▲	▲	▲	
PATHCOM															▲								
PORT	▲	▲	▲	▲	▲	▲		▲		▲	▲	▲	▲		▲	▲	▲	▲	▲	▲	▲	▲	
POSTCM																					▲		
PTR						▲																	
QUEUE	▲			▲						▲		▲											
RCCCOM									▲														
RCCOM									▲														
REFCOM																				▲			
REPORT	▲	▲	▲													▲	▲				▲		
RISKCOM					▲	▲	▲																
RLCOM																					▲		
SAD	▲			▲	▲	▲		▲		▲	▲	▲			▲	▲						▲	
SCHED	▲															▲	▲	▲	▲	▲		▲	
SETUP													▲										
SHIP	▲	▲	▲	▲	▲	▲		▲		▲	▲	▲	▲		▲	▲	▲	▲	▲	▲	▲		
SKED															▲								
SOLV									▲														
SREPT																					▲		
STOW	▲	▲	▲	▲				▲		▲	▲	▲	▲		▲	▲	▲	▲		▲	▲	▲	
TBASIS									▲														
TRANSP	▲							▲	▲	▲	▲	▲	▲		▲		▲			▲		▲	
VECTOR									▲			▲											
WARCS													▲	▲									

▲ Indicates the COMMON is used in the module

APPENDIX J

TABLE OF SUBROUTINE INTERACTIONS

MODULE	ROUTINES IN MODULE	CALLED BY MODULE
ATTRIT	DECOMP HYPERB INIRSK LNKRSK RISKIT RISKS	DECOMP HYPERB RISKS LNKRSK
BARK	DEBARK EMBARK NXCMT NXCP NXCSQF	NXCMT NXCSQF NXCP OSCOMP SYSRCS PORTTM WREXCL
BUNKER	FILL FUELS PITSTP	DISTIM FILL PITSTP
CHANEL	CHANEL CHORDR CLUSTR CULL ELIM FINDCP FINDIN INITCH INTERV PCLUS PODQ POEQ SPLIT TIMSRT	FNTIME WREXCL FINDIN FINDCP INITCH SYSRCS PCLUS CULL CLUSTR TIMSRT
CYCLE	ACCEPT BLDSOL CHEVAL CHINIT ENRICH EVLATE OPTION PORD QWAIT REJECT ROWGET SECON1 TEST WOODEN REAL FUNCTIONS: LODTIM RISK2 UNLTIM URAND	DISTIM QWAIT ACCEPT PORD REJECT EVLATE OPTION TEST SYSRCS ENRICH
DIST1	ADPORT ARC AVOID CANFND CASE1 CROUTE DISCHK GETDIS INIDIS INLINE INLINK INNODE NEAR SAD1 SADTAB SAVDIS SPA TABLED TOPOES LOGICAL FUNCTION: NTRSEX REAL FUNCTION: UNTIME	NEAR SYSRCS ARC DISCHK CANFND SPA CROUTE
DIST2	COMNOD DISCAL DISCPY DISFIL INVIEW KEEPRT PTOP SADDUN SADS SHPRTE SNEAR STAR	INVIEW GETDIS INNODE INLINK INLINE ADPORT INIDIS INIRSK AVOID STAR DISFIL SADS STAR DISFIL DISCPY INIRSK AVOID STAR SADS SAVDIS TABLED SADTAB SAD1 SYSRCS SPA PTOP RISKIT KEEPRT HYPERB SADDUN SNEAR NEAR CASE1 TOPOES SHPRTE ARC KEEPRT COMNOD FNTIME

MODULE	ROUTINES IN MODULE	CALLED BY MODULE
INPUT1	INCGO PORCHK	SYSRCS PORCHK WREXCL
INPUT2	FILLCP INPARA INPORT INSHIP INSTOW INTHRU MISMAT NITIAL	SYSRCS FILLCP
INRISK		INRISK
LCINPT	LCINPT	SYSRCS
LCROOT	LCYCLE LPRIME LSAILM LSTASH	MATGEN LPRIME LSAILM MATREV BLDSOL ROWGET LSTASH DISTIM SYSRCS
LP84	ARRDEP GNC GNCDAT GNCOUT SHRANK REAL FUNCTION: GTCOST	ARRDEP SHRANK GNCDAT MGNC GNCOUT SYSRCS
MAIN	COLPTR DELETE REDSOL RTLOAD1 RTLOAD2 WRTSOL	INPARA INSHIP INPORT INTHRU INSTOW NITIAL MISMAT INCGO DISCAL ELIM SPLIT INTERV PODQ CHANEL CHORDR POEQ RTLOAD1 RTLOAD2 GNC SIMOVR POST SYSRCS REDSOL NCYCLE WRTSOL SIMULA SAILIN ADEQAT DELETE
MATGEN	MATGEN MATREV INTEGER FUNCTION: TPCOST LOGICAL FUNCTION: OKNEO	SYSRCS COLPTR DISTIM
NOPT	NCYCLE NPRIME NSAILM NXTBST PVISIT STASH TEMPLD LOGICAL FUNCTION: FULL	MATGEN NPRIME NSAILM MATREV BLDSOL TEMPLD ROWGET CHINIT STASH SECON1 NXTBST CHEVAL DISTIM SYSRCS PVISIT
OPTA	ACYCLE ADDCHN APRIME ASAILM COLPTR COLSET COLSOL DEPCHK DONE PURGE	MATGEN FIRSTX SOLVE APRIME ASAILM WOODEN MATREV COLPTR ROWGET DONE BLDSOL SYSRCS CHINIT COLSET ADDCHN SECON1 CHEVAL DISTIM PURGE COLSOL

MODULE	ROUTINES IN MODULE	CALLED BY MODULE
PATH	CHKAMO CHKCH EXCH PATH PATHD REPATH SKEDIN STOPS SWPR	SYSRCS CENTER CHECK FAREST SRTULL STOPS PATHD SWPR CHKCH CHKAMO EXCH DISTIM
POST	CARSTO HISTO LCREP POST PWRREP REPTC REPTP REPTS REPTSC RLREPT STATS	LCREP REPTS STATS CARSTO HISTO REPTC RLREPT REPTS REPTSC REPTP PWRREP SYSRCS
REFINE	COMPAT DUMSHP FRACLD INIMRK NEWROW REFIN1 REFIN2 INTEGER FUNCTION: LV	DUMSHP INIMRK FRACLD SYSRCS NEWROW COMPAT INXTRA
REUSE	GNC2 MATCH RESEED SCANBAD SEEDAT SEEDCST SEEDSOL SORTL TEMPLD TRIAL	SEEDCST SEEDAT GNC2 SEEDSOL SCANBAD DISTIM TRIAL SORTL TEMPLD SYSRCS MGNC MATCH
S1	CHKCAP RISKCS SIMOVR SIMULA SLPCHK	SAILIN SLPCHK SIMULA REFIN1 ADEQAT SLPCHK REFIN2 SIMULA SKEDIN INXTRA ASSURE PATH REPATH NOSTOP FUELS ENDSIM EMBARK WREXCL STATUS DEBARK DISTIM RISKCS CHKCAP FINDCP
S1RE	ANYLAT CHKCAP FREETR RISKCS SIMOVR SIMULA SLPCHK	SIMULA REFIN1 ADEQAT FREETR ANYLAT SLPCHK REFIN2 SIMULA SAILIN SKEDIN INXTRA ASSURE PATH REPATH NOSTOP FUELS ENDSIM EMBARK WREXCL STATUS DEBARK DISTIM RISKCS SYSRCS CHKCAP FINDCP
S2	ALDCHK ASSURE CENTER CHANGE CHECK ENDSIM ENUF FAREST INSERT NOSTOP PORTTM REMOVE SRTULL STATUS UNDEL	SYSRCS FAREST DISTIM WREXCL UNDEL CHANGE ENUF REMOVE INSERT FINDCP
SOLVE	FIRSTX RCCOL RCGEN2 RCROW SOLVE SORTOF SORTRC	SYSRCS RCGEN2 RCROW RCCOL SORTOF
STOP	SYSRCS	

MODULE	ROUTINES IN MODULE	CALLED BY MODULE
UTIL	ADEQAT DISTIM FINDCP FNTIME INXTRA OSCOMP SAILIN SOLU WREXCL INTEGER FUNCTIONS: ANYCAP DISBAS DIST GCD LOGICAL FUNCTIONS: COPEN INFEAS REAL FUNCTIONS: RISKY RISTIM	SYSRCS FNTIME INXTRA

INTERNAL DISTRIBUTION

1. D. T. Bain
2. M. S. Bronzini
3-4. R. G. Edwards
5. P. S. Gillis
6. I. G. Harrison
7-16. J. E. Hawthorne
17-18. B. K. Jones
19. A. J. Klein
20. R. D. Kraemer
21-30. R. A. McLaren
31. I. R. Moisson
32. D. H. Pike
33. D. E. Reichle
34. R. B. Shellton
35. G. E. Whitesides/R. P. Leinius
36. E. W. Whitfield
37. K. H. Zimmerman
38. P. B. Zuschneid
39. Central Research Library
40. ORNL Y-12 Technical Library
41. Document Reference Section
42-44. Laboratory Records Department
45. Laboratory Records Department, RC
46. ORNL Patent Office

EXTERNAL DISTRIBUTION

47. Dr. Bruce G. Buchanan, Computer Science Department, University of Pittsburgh, 206 Mineral Industries Building, Pittburgh, PA 15260
48. Mr. John J. Cuttica, Vice President, End Use, Research and Development, Gas Reserch Institute, 8600 W. Bryn Mawr Avenue, Chicago, IL 60631
49. Dr. Allan Hirsch, President, Dynamac Corporation, The Dynamac Building, 11140 Rockville Pike, Rockville, MD 20852
50. Dr. Denton E. Morrison, 333 Oxford Road, East Lansing, MI 48823
51. Dr. Martin Williams, Professor, Department of Economics, Northern Illinois University, DeKalb, IL 60115
52. D. Alvic, Environment and Resources Center, The University of Tennessee, 327 Stadium Hall, Knoxville, TN 37996

53-62. P. Daley, Navy Regional Data Automation Center, Washington, Washington Navy Yard, Washington, DC 20374

63-72. H. Stout, Military Sealift Command, N6, Washington, DC 20398-5100

73-77. J. K. Wiswesser, Planning Support Division, Military Sealift Command, Washington, DC 20398-5100

78-82. Col R. D. Hunter, Transportation Branch/TCJ5-DT, U.S. Transportation Command, Building 1961, Scott AFB, IL 62225-7001

83. Office of Assistant Manager for Energy Research and Development, DOE/ORO, P. O. Box 2001, Oak Ridge, TN 37831-8600

84-93. Office of Scientific and Technical Information, Department of Energy, P. O. Box 62, Oak Ridge, TN 37831