



The Capacity of The Air Force Satellite
Control Network

THESIS

Kwangho Jang

Captain, Republic of Korea Army

AFIT/GSO/ENS/96D-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

DTIC QUALITY INSPECTED 3

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A

Approved for public release

19970205 031

AFIT/GSO/ENS/96D-01

The Capacity of The Air Force Satellite
Control Network

THESIS

Kwangho Jang
Captain, Republic of Korea Army

AFIT/GSO/ENS/96D-01

DTIC QUALITY INSPECTED 3

Approved for public release; distribution unlimited

This view expressed in this thesis are those of the author and do not reflect the official policy or position of the Department or the U.S. Government.

THESIS APPROVAL

Student : Kwangho Jang, Captain, ROKA

Class : GSO-96D

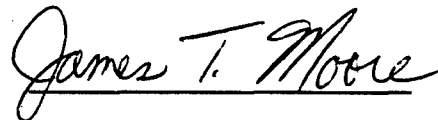
Title : The Capacity of the Air Force Satellite Control Network

Defense Date : 19 November 1996

Committee : Name/Title/Department

Signature

Advisor James T. Moore, Lt Col, USAF
Associate Professor
Department of Operational Sciences

Handwritten signature of James T. Moore in cursive script, underlined.

Reader William Paul Murdock, Major, USAF
Assistant Professor
Department of Operational Sciences

Handwritten signature of William Paul Murdock in cursive script, underlined.

AFIT/GSO/ENS/96D-01

The Capacity of The Air Force Satellite Control Network

THESIS

Presented to the Faculty of the Graduate School of Engineering
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Space Operations

Kwangho Jang, B.S.

Captain, ROKA

DECEMBER, 1996

Approved for public release; distribution unlimited

Preface

Determining the capacity of the Air Force Satellite Control Network (AFSCN) is a complex process. I obtained invaluable advice and information in understanding how sample data sets are generated. I would like to thank all my classmates of GSO-95D and GSO-96D for all their help and encouragement throughout the entire AFIT program.

I would also like to especially thank Lieutenant Colonel James Moore for his patience, insight, and guidance.

I also wish to thank my reader, Major William Paul Murdock.

I must thank my wife, Kyoungmin, and my son, Jun, for everything they have done. My family held me together throughout my entire stay in U.S.A. Their love and support was given without my asking. And of course, I must thank my parents for the faith they maintained in me despite my efforts to destroy it.

Finally, I would like to thank my Lord for his love. All I do is for his glory.

Kwangho Jang

Table of Contents

	Page
Preface.....	ii
List of Figures	vi
List of Tables	vii
Abstract.....	ix
I. Introduction.....	1-1
Overview.....	1-1
Background	1-1
Satellite.	1-2
Mission Control Complex.....	1-2
Remote Tracking Station	1-3
Resource Control Complex.....	1-3
Satellite Range Scheduling Process	1-4
Problem Statement	1-6
Research Objective	1-7
Overview of Thesis	1-7
II. Literature Review	2-1
Overview.....	2-1
Gooley's Research	2-1
Gooley's MIP Formulation	2-1
Decision Variables & Input Parameters.....	2-2
Gooley's Heuristic	2-3
Schalck's Research	2-5
Schalck's MIP Formulation	2-5
Schalck's Heuristic	2-7
Parish's Research	2-8
Genetic Algorithm	2-9
Simple Genetic Algorithm	2-9
Coding.....	2-9
Evaluation and Selection.....	2-9

Crossover Operator	2-9
Solution Methodology for SRS.....	2-10
Small Problem.....	2-10
Scheduling as a Sequencing Problem	2-11
Parish's Conclusions.....	2-13
III. Approach	3-1
Overview.....	3-1
Generating sample data sets for the low altitude satellite support requests.....	3-1
Analyze the Data.....	3-1
Obtaining the desired information	3-2
Procedure for Generation of the Sample data for low altitude satellite support requests	3-5
Input parameters for the procedure	3-5
Generating sample data sets for the medium and high altitude satellite support requests	3-10
Analyze the Data.....	3-10
Obtaining the desired information	3-10
Procedure for Generation of the Sample data for medium and high altitude satellite support requests	3-16
Input parameters for the procedure	3-16
Schedule and Analyze the Sample data sets using Schalck's SRS scheduling	
Algorithm.....	3-21
IV. Results.....	4-1
Overview.....	4-1
Sample data sets Generation	4-1
Results of generation of the low altitude satellite support requests sample data sets.....	4-1
Results of generation of the medium/high altitude satellite support requests sample data sets	4-3
Sorting the Data Sets.....	4-5
Scheduling Results.....	4-6
Summary of Results	4-29
V. Conclusions And Recommendations	5-1
Conclusions.....	5-1

Recommendations.....	5-2
Appendix A.	
<i>Low altitude satellite support requests sample data sets generation code.</i>	A-1
<i>Medium and High altitude satellite support requests sample data sets generation code.....</i>	A-4
Appendix B.	
<i>Sort Program.</i>	B-1
Appendix C.	
<i>Schalck's SRS algorithm programs.</i>	C-1
Appendix D.	
<i>The Results from Random Sample Generation.</i>	D-1
Appendix E.	
<i>The Results from Scheduling.....</i>	E-1
Bibliography.....	BIB-1
Vita.....	VITA-1

List of Figures

Figure	Page
2.1 Gooley's Heuristic	2-4
2.2 Schalck's SRS Scheduling Algorithm	2-8
2.3 Small Schedule.....	2-11
3.1 The Number of Supports(Low Altitude) visible to each RTS within 7 Day Period	3-3
3.2 The Number of Low Altitude Satellite Supports by Support Length (7 day period).	3-5
3.3 Flow Chart for Generating Low Altitude Satellite Support Requests.....	3-6
3.4 Flow Chart for Generating Medium and High Altitude Satellite Support Requests.	3-17
3.5 Determine the Capacity of AFSCN	3-22
4.1 The Scheduling Performance for Low Altitude Satellite Support Requests.....	4-29
4.2 The Scheduling Performance for Medium and High Altitude Satellite Support Requests when the Number of Low Altitude Satellite Support Requests is around 145.	4-30
4.3 The Scheduling Performance for Medium and High Altitude Satellite Support Requests when the Number of Low Altitude Satellite Support Requests is increased by 17.5%.	4-31
4.4 The Scheduling Performance for Medium and High Altitude Satellite Support Requests when the Number of Low Altitude Satellite Support Requests is increased by 35%.	4-31

List of Tables

Table	Page
2.1 Small Schedule Time Windows.....	2-10
2.2 Small Problem Schedule.....	2-12
3.1 Sample Low Altitude Satellite Support Database.....	3-2
3.2 The Number of Supports (Low Altitude) Visible to each RTS (7 day period).....	3-3
3.3 The Number of Low Altitude Satellite Supports by Required Length (7 day period).....	3-4
3.4 The Number of Medium and High Altitude Satellite Support Requests at the Combination of Several RTSs (Total request number is 1148).....	3-12
3.5 The Number of Medium and High Altitude Satellite Supports by Required Length. (7 day period).....	3-13
3.6 The Sample Supports Number and Visibility Gap for Medium and High Altitude Satellite.	3-15
4.1 Sample Data Sets for Low Altitude Satellite Support Requests.	4-2
4.2 The Number of Daily Low Altitude Satellite Support Requests.....	4-3
4.3 Sample Data Sets for Medium and High Altitude Satellite Support..	4-4
4.4 The Number of Daily Medium/High Altitude Satellite Support Requests..	4-5
4.5 Scheduling Performance for the Low Altitude requests: Case 1.	4-8
4.6 Scheduling Performance for the Medium/High Altitude Requests: Case 1.	4-9
4.7 Scheduling Performance for All Satellite Support Requests: Case 1.	4-10
4.8 Scheduling Performance for the Low Altitude requests: Case 2	4-11
4.9 Scheduling Performance for the Medium/High Altitude Requests: Case 2.....	4-12
4.10 Scheduling Performance for All Satellite Support Requests: Case 2.	4-13
4.11 Scheduling Performance for the Low Altitude requests: Case 3	4-14

4.12 Scheduling Performance for the Medium/High Altitude Requests: Case 3.....	4-15
4.13 Scheduling Performance for All Satellite Support Requests: Case 3.	4-16
4.14 Scheduling Performance for the Medium/High Altitude Requests: Case 4.....	4-17
4.15 Scheduling Performance for All Satellite Support Requests: Case 4.	4-18
4.16 Scheduling Performance for the Medium/High Altitude Requests: Case 5.....	4-19
4.17 Scheduling Performance for All Satellite Support Requests: Case 5..	4-20
4.18 Scheduling Performance for the Medium/High Altitude Requests: Case 6.....	4-21
4.19 Scheduling Performance for All Satellite Support Requests: Case 6..	4-22
4.20 Scheduling Performance for the Medium/High Altitude Requests: Case 7.....	4-23
4.21 Scheduling Performance for All Satellite Support Requests: Case 7..	4-24
4.22 Scheduling Performance for the Medium/High Altitude Requests: Case 8.....	4-25
4.23 Scheduling Performance for All Satellite Support Requests: Case 8..	4-26
4.24 Scheduling Performance for the Medium/High Altitude Requests: Case 9.....	4-27
4.25 Scheduling Performance for All Satellite Support Requests: Case 9..	4-28
4.26 Average scheduling Performance for All Requests..	4-32

Abstract

The daily mission objective of the Air Force Satellite Control Network (AFSCN) is to support communication with satellite systems. It is critical that the AFSCN operate 24 hours a day, seven days a week. Previous work on the satellite range scheduling problem has successfully scheduled over 90 percent of the satellite support requests. This research investigates the capacity of the AFSCN using an available satellite scheduling algorithm.

This research has three objectives. The first objective is to be able to generate sample data sets which represent a day of satellite support requests for low, medium, and high altitude satellites. The second research objective is to schedule the satellite support requests in the sample data sets. The third objective is to determine an upper bound on the number of support requests which can be supported by the AFSCN.

Based on the reported results, the AFSCN is able to support around 175 low altitude satellite support requests and 250 medium/high altitude satellite support requests. At this level of demand, the scheduling algorithm is able to schedule approximately 90 percent of the satellite support requests.

THE CAPACITY OF THE AIR FORCE SATELLITE CONTROL NETWORK

I. INTRODUCTION

Overview

During the entire life of any satellite or military space system, from prelaunch checkout to on-orbit operations, there is a requirement for constant control, support, and direction of the satellite and its assigned mission. The Air Force maintains this critical operations capability through the Air Force Satellite Control Network (AFSCN).

Background

The AFSCN manages communications between satellites and remote tracking stations. The AFSCN consists of four major systems : 1) satellites, 2) Mission Control Complexes (MCCs), 3) remote tracking stations (RTSs), and 4) Resource Control Complexes (RCCs). A MCC contains the personnel that maintain operational control over the satellite and determine what commands should be transmitted to the satellite. Operational control entails planning for future satellite communications, ensuring current communication requirements are met and analyzing the data sent from the satellite. The RTSs are located throughout the world and contain the personnel that control the

equipment which communicates with the satellites. The RTSs provide a necessary link between the MCC and the satellite, enabling MCC personnel to control the satellite and receive information from it. The RCCs oversee the entire AFSCN and provide a centralized location where coordination between MCC and RTS personnel can take place.

Satellite. In general, AFSCN supported satellites may be categorized by the altitude (low, medium, or high) at which they orbit. *Low-altitude satellites* are characterized by near-polar orbits, with altitudes ranging from 100 to 200 nautical miles. Their operational lifetimes are short, the satellites have a short pass duration (defined as the amount of time a satellite is visible from a single point on the earth), and they require contact with an average of 1.5 remote tracking stations per revolution. *Medium-altitude satellites* generally have an orbital inclination of near 90 degrees, with altitudes ranging from 1,000 to 12,000 nautical miles. Medium altitude satellites have an expected pass duration of 0.33 to 11 hours and normally require one remote tracking station contact every two orbits. *High-altitude satellites* orbit at altitudes exceeding 12,000 nautical miles with RTS visibilities varying between eleven hours and continuous visibility. As the orbital altitude of a satellite increases, the amount of time the satellite is visible to a RTS generally increases. RTS visibility and support criticality determine how restrictive a satellite support is in the AFSCN.

Mission Control Complex. A Mission Control Complex (MCC) or Satellite Operations Center (SOPC) contains the personnel and equipment that maintain operational control of one or more satellite programs and is located at a resource control complex. Satellite programs are grouped together in MCCs. Programs with similar support requirements and functions are usually located in the same MCC. Each MCC is responsible for determining the support requirements for its satellites. A MCC will provide the following information to the RCC for each requested support: Greenwich Mean Time (GMT) and tolerance in which the support can be scheduled, length of time

required for the support, RTS visibility, RTS set-up time (called RTS turn-around time), and special requests for equipment. The period of time the satellite is visible to a RTS is commonly called a pass. Because of the short visibilities associated with low altitude satellites, each support is scheduled for the entire time the low altitude satellites are visible to a RTS. This is an important differentiation between low altitude satellites and all other satellites. No scheduling tolerance is provided by the MCC to the RCC for scheduling of low altitude satellites while the MCC does provide a scheduling tolerance for other satellites.

Remote Tracking Station. Remote tracking stations (RTSs) are located around the world and contain the personnel and equipment that communicate with the satellites. There are nine RTSs located throughout the world. Five of the RTSs can support two satellites simultaneously and one can support three satellites simultaneously. Each RTS has at least one antenna-equipment package which is called a RTS side. Each antenna can communicate with at most one satellite at a time. The total number of antenna sides is the limiting factor of the AFSCN. Because the antenna must track the satellite in order to communicate, satellites must be visible to the RTS for a satellite support to be scheduled. Communicating with a satellite includes down-linking satellite status telemetry, tracking the satellite, and sending commands to the satellite. Currently, there are sixteen RTS sides which can perform these three general functions in the AFSCN.

Resource Control Complex. The Resource Control Complex contains both the schedulers who generate the satellite support schedule and the MCC personnel who establish satellite support requirements. Conflicts exist when two support requests compete for the same resource or RTS side and cannot be scheduled at some other RTS side. When conflicts cannot be avoided, schedulers notify the appropriate MCCs to de-conflict the support requests among themselves. This process involves one or more MCCs changing one of the following: the time or tolerance for a support when RTS

visibility allows, the length of a support, or, when both the RTS and MCC agree, the RTS setup time before a support.

Satellite Range Scheduling Process. Schedule construction for each day is started about two weeks before implementation. The requests for supports generated by a MCC for each satellite are listed on a program action plan (PAP) and received by the schedulers at the RCC. The requests allow the schedulers to begin building a feasible schedule and identify problem requests. As they become known, RTS personnel will provide downtime information and MCCs controlling low altitude satellites will provide support requests. MCCs controlling low altitude satellite programs generally do not submit PAPs because visibilities change too much to predict support times accurately until approximately 24 hours prior to the required support. For low altitude satellite programs, MCCs submit support requests anytime during the 24 hours prior to the requested support time.

In general, the schedulers tentatively schedule the relatively flexible PAP requests first, and insert the more restrictive low altitude satellite supports and protected RTS downtimes into the schedule as they become known. The low altitude satellite supports and protected downtimes almost always take priority over the medium and high altitude satellite support requests and unprotected downtimes. The schedule does not become firm until these low altitude satellite requests are received and scheduled by the RCC schedulers.

The schedule generation process can be divided into four separate phases: 1) generation of a "seven day" schedule, 2) generation of an initial 24 hour schedule, 3) conflict resolution, and 4) real-time scheduling.

The first phase in generating a daily schedule is collecting all the support requests for the week-long period beginning two weeks later. These requests are received via PAPs from the MCCs supporting high and medium altitude satellites. These relatively long-range requests are randomly scheduled around the RTS-requested downtimes

creating a tentative schedule which is distributed back through the AFSCN one week in advance. This schedule is called a seven day schedule and is not firm because the more restrictive low altitude satellite supports are not included. The schedule is used by AFSCN components to ensure all requests have been received and acknowledged by the RCC.

After the seven day schedule has been published, the schedule will be updated iteratively as requests for low altitude satellite supports and RTS downtimes are received to produce an initial 24 hour schedule. Scheduling priority generally reflects the flexibility associated with each request. The more restrictive low altitude satellite support requests usually take precedence over the more flexible medium and high altitude satellite support requests. However, a medium or high altitude satellite support request can take scheduling priority if the satellite support is critical to the satellite's mission.

RTS downtimes can be classified as 1) routine maintenance or 2) major maintenance/modification. Routine maintenance is generally quite flexible and the requirements are known in advance. These downtimes are handled much like a high altitude satellite request and are often re-scheduled in order to schedule a more restrictive, higher priority request. Major maintenance or modification includes equipment malfunction and often is short notice with little flexibility. The scheduling of this latter downtime type ranges from low altitude satellite-like scheduling to real-time schedule changes. As these downtimes are realized, they are incorporated into the schedule.

The conflict resolution phase is the 24 hour period prior to real-time and begins with the process of deconflicting the initial 24 hour schedule. All support and downtime requests have been scheduled and conflicts have been identified in the initial 24 hour schedule. The schedulers call the MCC personnel and/or RTS personnel involved in each conflict; options may be identified and the personnel will be expected to resolve the conflict by changing one of the following: the time or tolerance for a support if RTS

visibility allows, the length of a support, or the RTS setup time before a support if the RTS and MCC agree. If the conflict cannot be resolved in this manner, a support request will be canceled. In any conflict situation, all involved personnel will be notified and are expected to resolve the conflict.

Real-time scheduling occurs after a conflict-free 24 hour schedule is published and becomes the official schedule for the present day. Changes to this schedule would include satellite vehicle problems, RTS problems, changing mission requirements, incorrect requests by a MCC, or incorrect processing of a request by the schedulers. Changes to the published 24 hour schedule can affect up to one third of the scheduled requests during this phase. Changes to the schedule follow a formal priority system. By following this system, schedulers can quickly and efficiently determine which support request takes priority when a real-time change occurs and causes a conflict.

Problem Statement

Air Force and industry analysts often use computer models to study large scale scheduling. One particular scheduling problem of interest to the AFSCN is the Satellite Range Scheduling (SRS) problem. The AFSCN provides at least 300 command and control communications per day between nine remote tracking stations (RTSs) and approximately 100 satellites (2). Each communication between a satellite and a RTS is called a support.

In analyzing the capacity of the AFSCN, we must ask: what is the maximum number of satellite supports the AFSCN can accommodate?

The fundamental problem of determining the capacity of the AFSCN conducting 24-hour satellite support operations is threefold. The first problem is finding cyclical sample data that meet the current request conditions of the satellite supports. The second problem is solving the satellite range scheduling (SRS) problem for the sample data sets.

The third problem is analyzing the solution of the SRS and determining the capacity of the AFSCN.

Research Objective

This research is designed to satisfy three objectives. The first objective is to be able to generate sample data sets which represent a day of satellite support requests for the low, medium, and high altitude satellites. The second research objective is to schedule the satellite support requests in the sample data sets. The third objective is to determine the maximum number of supports which can be supported by the AFSCN.

Overview of Thesis

This thesis is organized into five chapters. Chapter 2 reviews the current literature related to satellite support requests scheduling. Chapter 3 presents the methodology developed to accomplish the research objectives. The results are presented and discussed in Chapter 4, and Chapter 5 presents conclusions and offers some recommendations for further research.

II. LITERATURE REVIEW

Overview

This chapter discusses the fundamental theory and the information that apply to the problem of estimating the capacity of the AFSCN. It begins with previous studies of the AFSCN satellite range scheduling (SRS) problem.

Gooley's research

Gooley (2) creates an algorithm for the SRS problem that successfully schedules around 90 percent of the satellite support requests. He developed a two phased approach. The first phase scheduled as many low altitude satellite supports as possible, while the second phase scheduled as many additional medium and high altitude satellite supports as possible. For both phases, schedule generation and schedule improvement algorithms were developed. For low altitude satellites, the schedule generation algorithm applied a mixed integer program (MIP) with a linking procedure, and the schedule improvement algorithm was a two satellite interchange procedure. For medium/high altitude satellites, the schedule generation algorithm was an insertion procedure and the schedule improvement algorithm was a three satellite interchange procedure.

Gooley's MIP Formulation. A MIP problem is a linear programming (LP) problem in which some of the variables are required to be non-negative integers.

Decision Variables & Input Parameters. The decision variables and input parameters of Gooley's MIP are defined below.

Parameters and sets:

BV_{ij} - Beginning of visibility window for support *i* at RTS *j*.

EV_{ij} - Ending of visibility window for support *i* at RTS *j*.

R_{ij} - Length of requested support *i* at RTS *j*.

TO_i - Turnaround time for support *i*.

SDT_j - Start of downtime for RTS *j*.

EDT_j - End of downtime for RTS *j*.

RTS_i - represents the set of RTSs to which support *i* is visible

M - large positive constant value

Decision variables:

ST_{ij} - Start time of support *i* at RTS antenna *j*

$$X_{ij} = \begin{cases} 1 & \text{if support } i \text{ is scheduled at RTS side } j \quad \forall i, j \in \text{RTS}_i \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ihj} = \begin{cases} 1 & \text{if } ST_{hj} < ST_{ij} \quad i \neq h \\ 0 & \text{if } ST_{hj} \geq ST_{ij} \end{cases}$$

Objective Function:

$$\text{Maximize } \sum_{i=1}^n \sum_{j \in \text{RTS}_i} X_{ij} \quad (2-1)$$

Subject to:

$$\sum_{j \in RTS_i} X_{ij} \leq 1 \quad i = 1 \dots n \quad (2-2)$$

$$ST_{ij} \geq BV_{ij} * X_{ij} \quad i = 1 \dots n, \quad \forall j \in RTS_i \quad (2-3)$$

$$ST_{ij} \leq (EV_{ij} - R_{ij}) * X_{ij} \quad i = 1 \dots n, \quad \forall j \in RTS_i \quad (2-4)$$

$$ST_{ij} + R_{ij} \leq SDT_{hj} + M(1 - X_{ij}) \quad \forall i \quad \text{given } j \quad (2-5)$$

$$ST_{ij} \geq EDT_j + TO_i - M(1 - X_{ij}) \quad \forall i \quad \text{given } j \quad (2-6)$$

$$ST_{ij} + R_{ij} + TO_h \leq ST_{hj} + My_{ihj} + M(1 - X_{ij}) + M(1 - X_{hj}) \quad \forall (i, h) \quad \text{given } j \quad (2-7)$$

$$ST_{hj} + R_{hj} + TO_i \leq ST_{ij} + M(1 - y_{ihj}) + M(1 - X_{ij}) + M(1 - X_{hj}) \quad \forall (i, h) \quad \text{given } j \quad (2-8)$$

$$X_{ij} \in \{0,1\}$$

$$y_{ihj} \in \{0,1\}$$

$$ST_{ij} \geq 0 \quad \forall i, j \in RTS_i$$

Gooley's Heuristic (2:4-1). Gooley's scheduling heuristic generates a schedule and then attempts to improve the schedule. The heuristic consists of two phases: the first phase schedules as many low altitude satellite supports as possible, while the second phase schedules as many medium and high altitude satellite supports as possible. In each phase, a schedule generation algorithm schedules the satellite supports. Then a schedule

improvement algorithm adjusts the schedule so that unscheduled satellite supports can be added to the schedule. The MIP linking procedure and a satellite insertion procedure are the schedule generation algorithms. The MIP linking procedure schedules the low altitude satellite supports. Then a satellite insertion procedure schedules medium/high altitude satellite supports. The two satellite interchange procedure and the three satellite interchange procedure are part of the schedule improvement algorithm.

Figure 2.1 shows the flow of Gooley's heuristic approach (2:4-2).

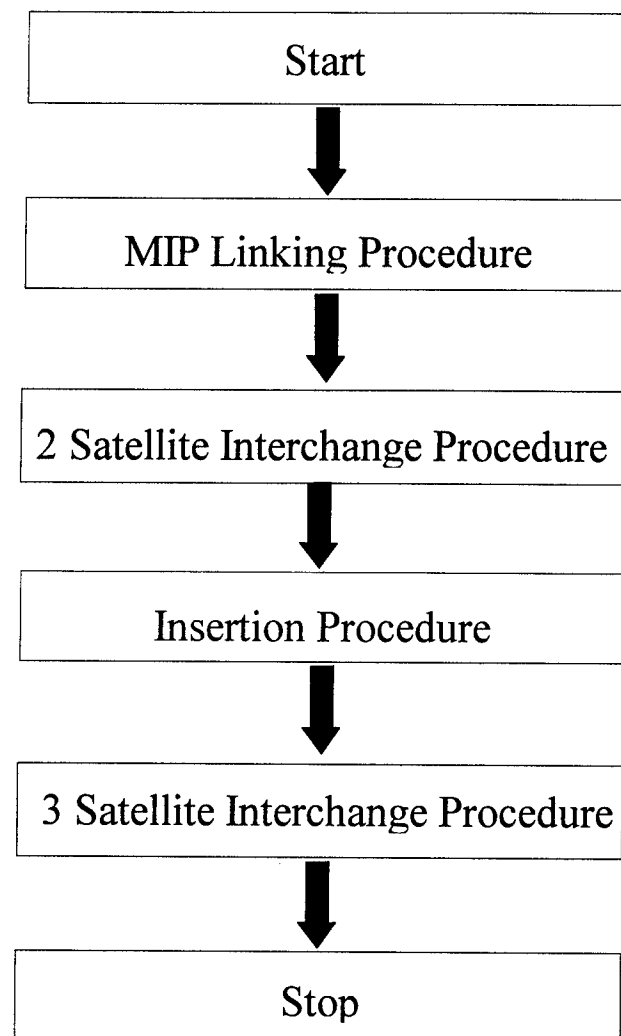


Figure 2.1 Gooley's Heuristic

The results from the algorithm were very encouraging. Almost 91% of all satellite support requests were scheduled, but there are some limitations in this approach. The scheduling of special case supports was not attempted. Examples of special case supports are a high-priority support request, an interval support request, and a simultaneous support request. Each of these cases has more complicated support requirements. The high-priority request requires service by more than one RTS antenna. The interval request is where a satellite must have a support every "X" hours. The simultaneous request is where a satellite communicates with two RTS antennas simultaneously.

Schalck's research

Schalck (3) improved on Gooley's approach by reducing the number of integer variables needed in the MIP formulation. An iterative heuristic approach was used to schedule satellite support requests in three successive subproblems. The first subproblem involves scheduling low altitude satellite support requests using a mixed integer programming approach. Each of the next two subproblems involves scheduling 12 hour blocks of medium and high altitude satellite support requests, again using a MIP approach.

Schalck's MIP Formulation.

Parameters and sets:

BV_{ij} - Beginning of low altitude satellite visibility window or medium or high altitude satellite tolerance window for support *i* at RTS *j*.

EV_{ij} - Ending of low altitude satellite visibility window or medium or high altitude satellite tolerance window for support *i* at RTS *j*.

R_{ij} - Length of requested support *i* at RTS *j*.

TO_i - RTS set-up or turn-around time for support *i*.

RTS_i - set of RTS sides where support i is feasible.

M - large positive constant value.

D12 - set of overlapping support request combinations at a particular RTS side j where both requests can be feasibly scheduled only with support i before support h at j .

D21 - set of overlapping support request combinations at a particular RTS side j where both requests can be feasibly scheduled only with support h before support i at j .

DE - set of overlapping support requests combinations at a particular RTS side j where both requests can be feasibly scheduled in either order at j .

DN - set of overlapping support requests combinations at a particular RTS side j where both requests cannot be feasibly scheduled at j .

Decision variables:

ST_{ij} - Start time of support i at RTS antenna j

$$X_{ij} = \begin{cases} 1 & \text{if support } i \text{ is scheduled at RTS side } j \quad \forall i, j \in RTS_i \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ihj} = \begin{cases} 1 & \text{if } ST_{hj} < ST_{ij} \quad i \neq h \\ 0 & \text{if } ST_{hj} \geq ST_{ij} \end{cases}$$

Objective Function:

$$\text{Maximize } \sum_{i=1}^n \sum_{j \in RTS_i} X_{ij} \quad (2-9)$$

Subject to:

$$\sum_{j \in RTS_i} X_{ij} \leq 1 \quad i = 1 \dots n \quad (2-10)$$

$$ST_{ij} \geq BV_{ij} * X_{ij} \quad i = 1 \dots n, \quad \forall j \in RTS_i \quad (2-11)$$

$$ST_{ij} \leq (EV_{ij} - R_{ij}) * X_{ij} \quad i = 1 \cdots n, \quad \forall j \in RTS_i \quad (2-12)$$

$$ST_{hj} - ST_{ij} + .5 \leq M(1 - y_{ihj}) \quad \forall (i, h, j) \in DE \quad (2-13)$$

$$ST_{ij} - ST_{hj} \leq M * y_{ihj} \quad \forall (i, h, j) \in DE \quad (2-14)$$

$$ST_{ij} + R_{ij} + TO_h \leq ST_{hj} + M * y_{ihj} + M(1 - X_{ij}) + M(1 - X_{hj}) \quad \forall (i, h, j) \in DE \quad (2-15)$$

$$ST_{hj} + R_{hj} + TO_i \leq ST_{ij} + M(1 - y_{ihj}) + M(1 - X_{ij}) + M(1 - X_{hj}) \quad \forall (i, h, j) \in DE \quad (2-16)$$

$$ST_{ij} + R_{ij} + TO_h \leq ST_{hj} + M(1 - X_{ij}) + M(1 - X_{hj}) \quad \forall (i, h, j) \in D12 \quad (2-17)$$

$$ST_{hj} + R_{hj} + TO_i \leq ST_{ij} + M(1 - X_{ij}) + M(1 - X_{hj}) \quad \forall (i, h, j) \in D21 \quad (2-18)$$

$$X_{ij} + X_{hj} \leq 1 \quad \forall (i, h, j) \in DN \quad (2-19)$$

$$X_{ij} \in \{0,1\}$$

$$y_{ihj} \in \{0,1\}$$

$$ST_{ij} \geq 0 \quad \forall i, j \in RTS_i$$

Schalck's Heuristic (3:3-8). Figure 2.2 outlines the heuristic approach developed by Schalck. The first step in the approach is to separate the requests into smaller sets with fewer requests. The smaller sets are scheduled iteratively one at a time, ensuring subsequent iterations do not change the schedule developed by earlier iterations. The smaller sets are scheduled using a MIP approach. Finally, the blocks of scheduled requests are combined into a 24 hour schedule. Implementing this approach requires a logical criteria for dividing the similar support requests into blocks.

The heuristic scheduled approximately 98% of the requested supports. These results do not include RTS down-times or overflow data-processing and display (DPAD) supports in the schedule. Because of the flexibility of the heuristic, any time block for a protected RTS down-time or special satellite support can be set aside before a particular schedule is created.

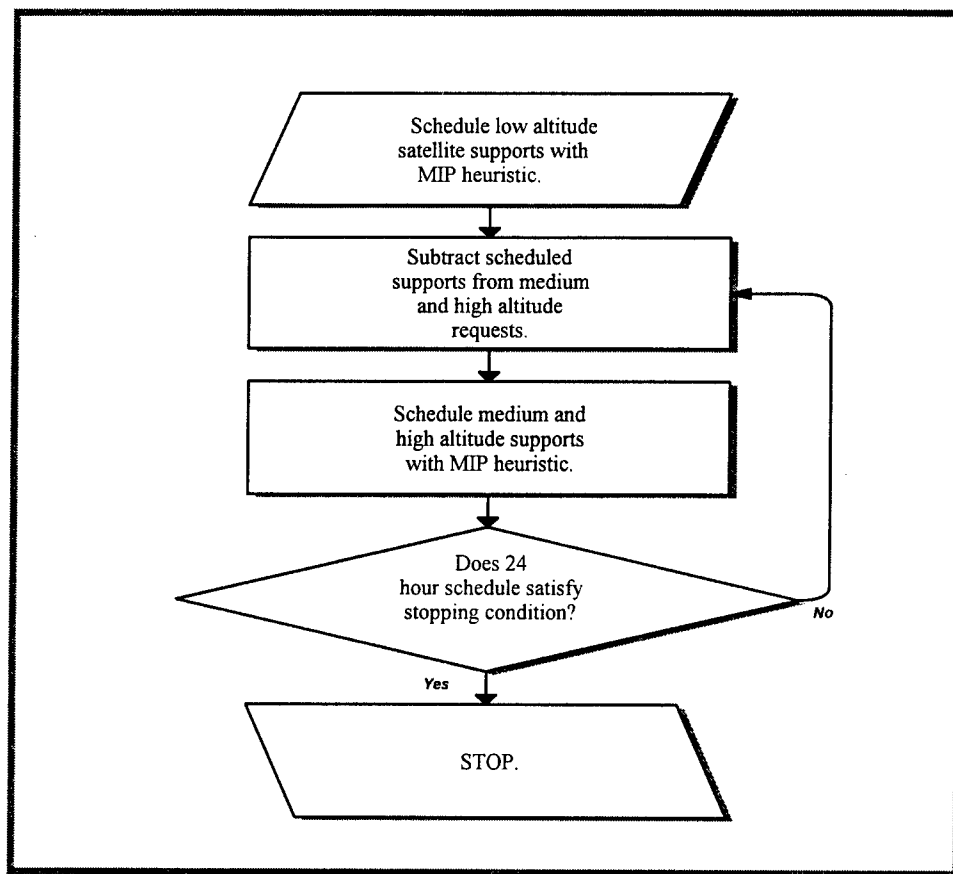


Figure 2.2 Schalck's SRS Scheduling Algorithm

Parish's research

Parish (4) used a genetic algorithm approach. The genetic algorithm approach attempted to find the best priority ordering of support requests, and then used a schedule builder program to build schedules based on simple rules. A schedule was produced for seven days of representative satellite range data with slightly better results compared to earlier efforts using a mixed-integer programming formulation. The main strength of genetic algorithms is their ability to quickly explore a large number of possible solutions for good, if not optimal, solutions.

This SRS scheduling approach used the genetic algorithm package GENITOR (5).

Genetic Algorithms. Genetic algorithms were developed by John Holland (6). His ideas were based on the biological theory of evolution where variations in chromosomes, or genetic codes, result in different traits in the individual.

Simple Genetic Algorithm. Most genetic algorithm work is based on the simple genetic algorithm as developed by Holland and described by Michalewicz (7).

Coding. The first, and generally most difficult, step of any genetic algorithm is to choose a proper coding to map the problem solution space into a genetic string, or chromosome, and to randomly create an initial population of individuals with varying strings. In the simple genetic algorithm, this coding is a binary string of zeros and ones. For function optimization, groups of binary digits are mapped so as to translate to a real number parameter representing the function value.

Evaluation and Selection. The strings in the population can be evaluated for their fitness relative to other strings in the population by entering their parameters into an evaluation or fitness function. The best strings reproduce by mating with each other to produce offspring for the next generation of the population. In the simple genetic algorithm, selection is governed by a "roulette wheel" selection operator. Each string has a probability of reproducing in proportion to the ratio of its fitness to the total fitness of the population. New strings are selected for reproduction by randomly "spinning" the wheel. The strings whose ratios are greatest should, on average, be selected more often than the less-fit strings which have small ratios.

Crossover Operator. During mating, two strings swap part of their "genetic" material. The children of these matings have parts of each of their parent's string. The swapping of genetic material, called crossover, allows for new child strings to be created,

combining good aspects of their parents. The resulting strings with above-average fitness tend to survive and prosper, while those with below-average fitness tend to die out.

Solution Methodology for Satellite Range Scheduling. By formulating the scheduling problem as a sequencing problem rather than as a mathematical program, a strait forward solution by an order-based genetic algorithm is possible.

Small Problem. An example set of time windows for a small set of five support requests is shown in Table 2.1. "Support" is an arbitrary support number, "Begin" is the starting time for a visibility window, "End" is the ending time for a visibility window, "Length" is the actual service time needed, and "TAT" is the setup time required before a service time can begin. These supports are all serviced by RTS "POGO-A".

Table 2.1 Small Schedule Time Windows.

Support	Begin	End	Length	TAT
1	1	13	3	1
2	15	22	3	2
3	7	17	3	1
4	1	10	3	1
5	2	8	3	2

This small problem is simplified since a real day's schedule would include over 300 supports, nearly all of which would have alternate windows for scheduling. These

alternate windows could come from different antennas or sides at the same RTS, or from different RTSs.

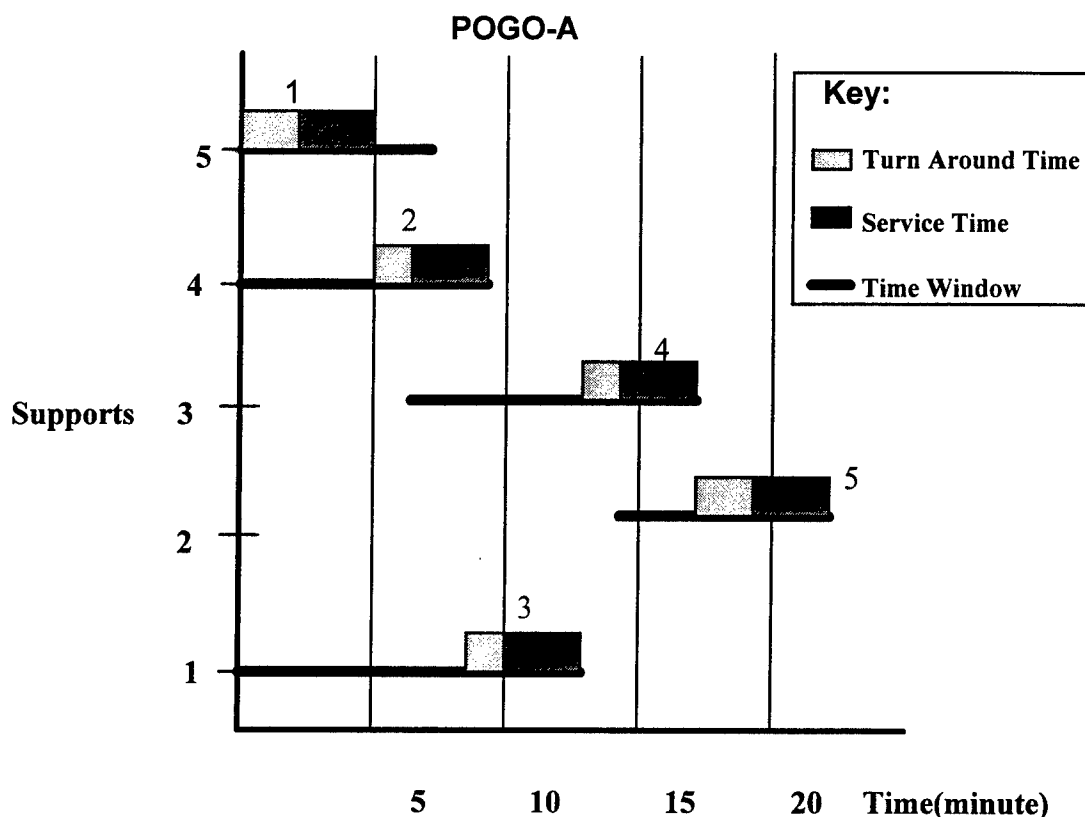


Figure 2.3 Small Schedule

In Figure 2.3, the horizontal axis shows time in minutes, the vertical axis shows the supports by support number, and the entire chart is for POGO-A. The time window is represented by a thin line, while the required TAT and service time is shown by the boxes. Such a representation clearly shows a schedule. The schedule for the example successfully schedules all supports with no overlap of the support times and no violation of time windows. The TAT for support 5 is outside the service support window, but this is legal as no communications with the satellite take place at this time.

Scheduling as a Sequencing Problem. Instead of formulating the satellite range scheduling problem as a mixed-integer program, Parish's research approaches the problem as a sequencing problem where a sequence is defined as an order in which to schedule items using simple rules for placing each item in the schedule. For example, in the SRS problem, the solution is represented as a sequence of supports. An example ordering is: (5, 4, 1, 3, 2).

In the example schedule, the first support, 5, would be placed in the first available time segment. The entire schedule is open and support 5 is scheduled at the beginning of its time window from time 2 to time 5. Note that the turn-around time of two minutes is scheduled from time 0 to time 2 which is outside the time window. This is valid since no communications take place during the turn-around time. The second support in the list is support 4. Its time window begins at time 1, but it cannot be placed there because support 5 has already used minutes 0 to 5. Support 4 is then scheduled from time 6 to time 9. This procedure repeats for supports (1, 3, 2), and completes a schedule as shown in Figure 2.3 and Table 2.2.

Table 2.2 Small Problem Schedule.

Support	Begin	End	Length	TAT
5	2	5	3	2
4	6	9	3	1
1	10	13	3	1
3	14	17	3	1
2	19	22	3	2

The actual start and end times for each support were determined by examining the completed schedule. Thus, all the information needed to define a solution is contained in the sequence of supports.

Genetic algorithms can quickly search the solution space, defined as the permutations of the list of items. These permutations can then be used to build a schedule as described above, and the number of supports scheduled successfully can be used as a fitness measure for the genetic algorithm.

Parish's Conclusions. Results, using one week of satellite support data, indicate a genetic algorithm can be successful in scheduling satellite supports. Results match or exceed those achieved by Schalck's MIP approach.

Parish's algorithm scheduled over 98% of requested supports (4). These results suggest a genetic algorithm (GA) based method can produce good schedules for the SRS problem.

III. APPROACH

This chapter discusses a methodology for estimating the capacity of the AFSCN. The methodology of this approach includes three parts. The first part is generating sample data sets of low altitude satellite support requests, the second part is generating sample data sets of the medium and high altitude satellite support requests, and the third part is scheduling the satellite support requests in the sample data sets using Schalck's satellite range scheduling algorithm (3).

Generating sample data sets for the low altitude satellite support requests

Analyze the data. This analysis is based on the data from Schalck's research (3). The SRS data used was the ASTRO general list database provided by Space Planning Flight, 21 SOPS, which lists all the information for each support (2). Two steps were required before this data was in the proper form for analysis: 1) separate the database into two databases with one for low altitude satellite supports and the other for medium and high altitude satellite supports, and 2) sort the low altitude satellite support database. Table 3.1 shows the format of a database with low altitude satellite supports. A support in this database is identified by an identification number (Ident) and revolution number combination and is given a support number for identification purposes. Additionally, each entry in the database associates a specific support with a specific RTS. In Table 3.1, the first column is the support number, the second column is the RTS, the third column is the starting visibility, the fourth column is the ending visibility, the fifth column is the request length, the sixth column is the turn around time, the seventh column is the identification number, and the last column is the revolution number.

Table 3.1 Sample Low Altitude Satellite Support Database.

Sup No.	RTS	BV	EV	Req Len	TAT	Ident	Rev No.
20	GUAM-A	212	229	17	20	1056	1445.4
20	GUAM-B	212	229	17	20	1056	1445.4
21	HULA-A	215	232	17	20	0286	4529.0
21	HULA-B	215	232	17	20	0286	4529.0
22	REEF-A	226	237	11	20	9757	2460.5

Now, the information from available data sets must be analyzed. The first step in this analysis is to determine the important characteristics of the data needed to create the data sets required to test the capacity of the AFSCN. For the low altitude satellite supports, the important characteristics are RTS, beginning visibility time, ending visibility time, and request length.

Obtaining the desired information. There are nine RTSs used by the AFSCN. Each RTS is visible to a certain number of low altitude satellites. A low altitude satellite support requirement is defined as a time window when an RTS is visible to a satellite and some form of communication is required. The data is assembled from each of the nine RTSs, and summarized as shown in Table 3.2 and Figure 3.1. In Table 3.2, the nine RTSs are listed in the first column. Columns D1 to D7 list by day 1 to day 7, the number of low altitude satellites which are visible to the RTS and require support by the RTSs listed in the first column. Figure 3.1 shows the number of supports each RTS is tasked with by day.

Table 3.2 The Number of Supports (Low Altitude) Visible to each RTS (7 day period).

	D1	D2	D3	D4	D5	D6	D7	TOTAL	Avg
BOSS	23	17	21	17	16	17	18	129	18.4286
COOK	11	8	12	8	6	8	10	63	9
GUAM	11	12	13	9	9	8	14	76	10.8571
HULA	20	13	13	17	15	14	18	110	15.7143
INDI	5	6	3	5	4	3	7	33	4.71429
LION	18	24	14	15	18	18	22	129	18.4286
PIKE	6	12	11	10	11	7	8	65	9.28571
POGO	49	37	46	53	54	57	36	332	47.4286
REEF	10	8	13	8	9	12	9	69	9.85714
TOTAL	153	137	146	142	142	144	142	1006	143.714

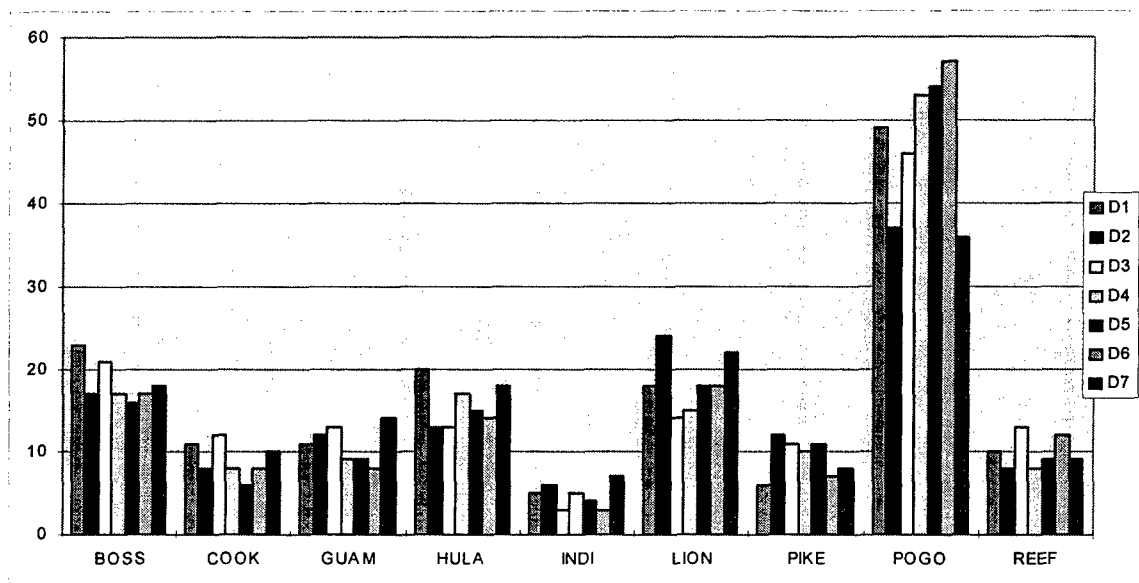


Figure 3.1 The Number of Supports (Low Altitude) Visible to each RTS within 7 Day Period.

Another key factor is the support length. The total support length is comprised of the length of the support requirement and the turnaround time for the RTS where the support is scheduled. The turnaround time is the amount of time required by a RTS to reconfigure for a support. The length of the support requirement depends on the type of satellite and the mission of the satellite. For low altitude satellite support requests, the support length is equal to the length of visibility.

**Table 3.3 The Number of Low Altitude Satellite Supports by Required Length
(7 day period).**

	D1	D2	D3	D4	D5	D6	D7	TOTAL	Avg
5min	0	0	1	0	0	0	1	2	0.285714
9min	1	1	1	1	0	1	2	7	1
10min	2	4	3	2	2	4	6	23	3.285714
11min	10	8	8	5	9	11	6	57	8.142857
12min	14	13	10	16	10	7	9	79	11.28571
13min	20	14	15	15	21	10	19	114	16.28571
14min	32	19	39	25	23	35	27	200	28.42857
15min	25	30	22	30	29	29	33	198	28.28571
16min	43	39	39	32	39	36	28	256	36.42857
17min	6	9	7	15	8	10	8	63	9
18min	0	0	0	0	1	0	0	1	0.142857
40min	0	0	0	1	0	0	0	1	0.142857
45min	0	0	1	0	0	1	2	4	0.571429
50min	0	0	0	0	0	0	1	1	0.142857
TOTAL	153	137	146	142	142	144	142	1006	143.7143

For a low altitude satellite, the support length ranges between five minutes and 50 minutes. Table 3.3 and Figure 3.2 show the number of low altitude satellite support requests by support length. Turn around time is not included in the support length.

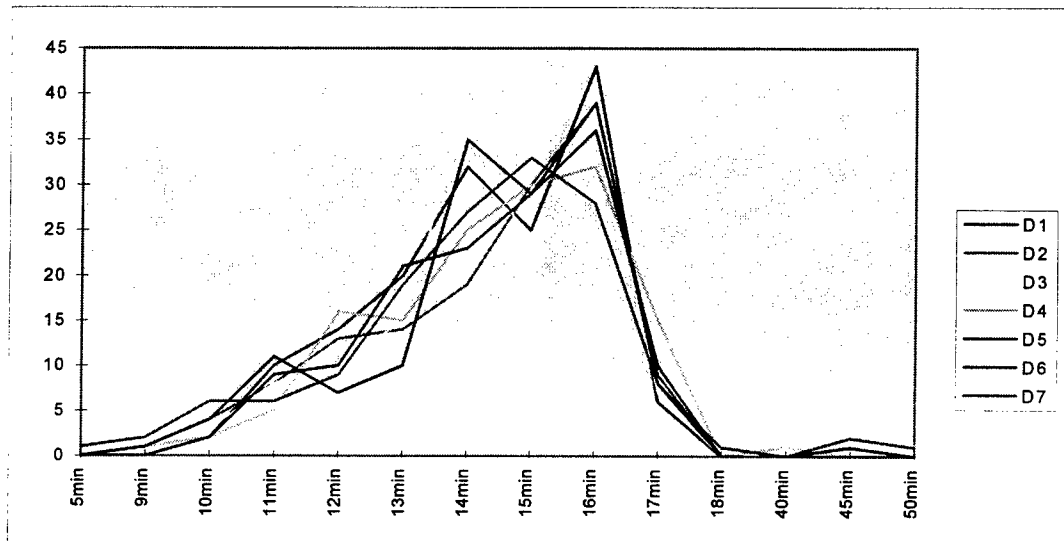


Figure 3.2 The Number of Low Altitude Satellite Supports by Support Length (7 day period).

Procedure for Generation of the Sample Data for Low Altitude Satellite Support Requests. The procedure presented below uses random number generation to create a sample data set of low altitude satellite support requests. On each pass through the procedure, a complete day of requests is created.

A flow chart of the procedure is presented in Figure 3.3 and the programs used are presented in Appendix A.

Input Parameters for the Procedure. The input parameters for the procedure are defined as:

I - index for support lengths.

J - index for RTSs.

STz[I] - particular length requested for I=1 to 14.

STn[I] - number of support requests of the particular length for I=1 to 14.

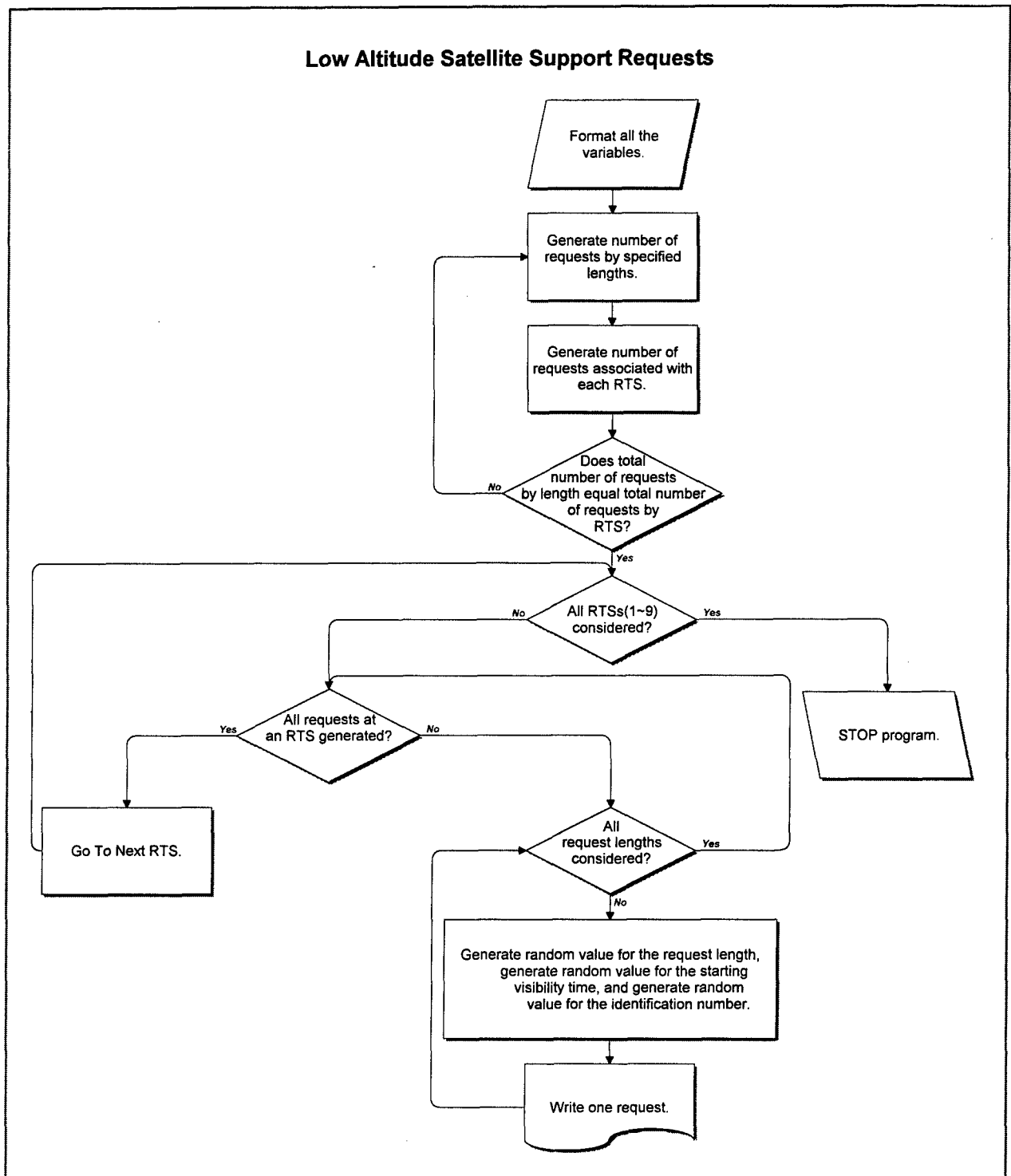


Figure 3.3 Flow Chart for Generating Low Altitude Satellite Support Requests.

RT[J] - name of RTS J for J=1 to 9.

RTn[J] - number of support requests at RTS J for J=1 to 9.

TM - total number of supports requested.

TR - total number of supports requested at the RTSs.

SV - starting time of low altitude satellite visibility window.

EV - ending time of low altitude satellite visibility window.

Step 0: (Initialization) Put STz[1] = 5, STz[2] = 9, STz[3] = 10, STz[4] = 11,
STz[5] = 12, STz[6] = 13, STz[7] = 14, STz[8] = 15, STz[9] = 16, STz[10] =
17, STz[11] = 18, STz[12] = 40, STz[13] = 45, STz[14] = 50, RT[1] = BOSS,
RT[2] = COOK, RT[3] = GUAM, RT[4] = HULA, RT[5] = INDI, RT[6]
= LION, RT[7] = PIKE, RT[8] = POGO, and RT[9] = REEF.

Step 1: (Generate random number STn[I])

Set STn[1] = round(random) for $0 \leq \text{STn}[1] \leq 1$,
STn[2] = round(2*random) for $0 \leq \text{STn}[2] \leq 2$,
STn[3] = round(4*random+2) for $2 \leq \text{STn}[3] \leq 6$,
STn[4] = round(6*random+5) for $5 \leq \text{STn}[4] \leq 11$,
STn[5] = round(9*random+7) for $7 \leq \text{STn}[5] \leq 16$,
STn[6] = round(11*random+10) for $10 \leq \text{STn}[6] \leq 21$,
STn[7] = round(20*random+19) for $19 \leq \text{STn}[7] \leq 39$,
STn[8] = round(11*random+22) for $22 \leq \text{STn}[8] \leq 33$,
STn[9] = round(15*random+28) for $28 \leq \text{STn}[9] \leq 43$,
STn[10] = round(9*random+6) for $6 \leq \text{STn}[10] \leq 15$,
STn[11] = round(random) for $0 \leq \text{STn}[11] \leq 1$,
STn[12] = round(random) for $0 \leq \text{STn}[12] \leq 1$,

$STn[13] = \text{round}(2 * \text{random})$ for $0 \leq STn[13] \leq 2$, and

$STn[14] = \text{round}(\text{random})$ for $0 \leq STn[14] \leq 1$.

$$TM = \sum_{I=1}^{14} STn[I].$$

Step 2: (Generate random number $RTn[J]$)

Set $RTn[1] = \text{round}(7 * \text{random} + 16)$ for $16 \leq RTn[1] \leq 23$,

$RTn[2] = \text{round}(4 * \text{random} + 8)$ for $8 \leq RTn[2] \leq 12$,

$RTn[3] = \text{round}(6 * \text{random} + 8)$ for $8 \leq RTn[3] \leq 14$,

$RTn[4] = \text{round}(7 * \text{random} + 13)$ for $13 \leq RTn[4] \leq 20$,

$RTn[5] = \text{round}(4 * \text{random} + 3)$ for $3 \leq RTn[5] \leq 7$,

$RTn[6] = \text{round}(10 * \text{random} + 14)$ for $14 \leq RTn[6] \leq 24$,

$RTn[7] = \text{round}(6 * \text{random} + 6)$ for $6 \leq RTn[7] \leq 12$,

$RTn[8] = \text{round}(21 * \text{random} + 36)$ for $36 \leq RTn[8] \leq 57$, and

$RTn[9] = \text{round}(5 * \text{random} + 8)$ for $8 \leq RTn[9] \leq 13$.

$$TR = \sum_{J=1}^9 RTn[J].$$

Step 3: (Backtracking) If $TR \neq TM$, go to Step 1; otherwise, go to Step 4.

Step 4: (Generating RTS) Select $RTn[J]$ and J is incremented by one for each repetition

from $J = 1$ to $J = 9$. If $J > 9$, go to Step 9; otherwise, go to Step 5.

Step 5: (Generating number of RTS) K is incremented by one for each repetition from

$K = 1$ to $K = RTn[J]$. If $K > RTn[J]$, go to Step 4; otherwise, go to Step 6.

Step 6: (Generating request length) Randomly select I for $1 \leq I \leq 14$. This gives a

particular length $STz[I]$. A Boolean comparison on a counter is done to determine if the number of allowable lengths, $STn[I]$, has been exceeded. If yes, repeat Step 6; otherwise set request length to $STz[I]$ and go to Step 7.

Step 7: (Generating SV and ID) Set $SV = \text{round}(\text{random} * 1440)$ and $ID = \text{round}(\text{random} * 8000 + 1000)$. If generated SV and ID are already used, then generate again. Set $EV = SV + STz[I]$.

Step 8: (Writing the Results) Write the information for the request to the output file. Then go to Step 5.

Step 9: (Termination) Stop the program.

Step 1 is designed to set the number of supports of length $STn[I]$. The random number is generated by a random number function. The random number function returns a real random number between 0 and 1. The range of the number of the $STn[I]$ is determined by analyzing the available data sets, as shown in Table 3.3. The lower bound is the lowest number of requests of the particular length observed over the seven day period while the upper bound is the highest number observed. After determining each $STn[I]$, sum these values and set equal to TM . At Step 2, $RTn[J]$ is generated in a similar way. TR is the sum of the $RTn[J]$. At Step 3, the number of satellite supports is a fixed number. If $TR \neq TM$, go to Step 1 and reaccomplish the generation. This step avoids the problem of the number of support requests not equaling the number of requests allocated to the RTSs. At Step 4, $RTn[J]$ is set to the number of supports at the J th RTS. There are nine RTSs. Therefore, J is incremented by one for each repetition of this Step until J equals nine. At Step 5, each RTS has a fixed number of satellite support requests. K is incremented by one for each repetition of this step until K reaches the total number of

satellite support requests at the designated RTS. At Step 6, the other key factor, which is request length is considered. The value I is generated randomly and used to designate a particular support length $ST_z[I]$. A check is also done to insure that $ST_n[I]$, the number of supports of a particular length, has not been exceeded. At Step 7, SV and ID are generated by using the random number function. Also, EV is set to the sum of SV and the support length determined at Step 6. At Step 8, the information from the previous steps needed to describe a satellite support request is written to a file. At Step 9, if all conditions at Step 4 are satisfied, the program stops. A sample data set is shown in Appendix D.

Generating sample data sets for the medium and high altitude satellite support requests.

Analyze the data. The medium and high altitude satellite support request data analyzed is based on the data from Schalck's research (3). The format of this database is the same format as that used for low altitude satellite support requests. Table 3.1 shows a sample of the format of the database.

Obtaining the desired information. Nine RTSs are also used for medium and high altitude satellite supports. There is a major difference between the low altitude and medium/high altitude satellite support requests. The low altitude satellite case has only one RTS per satellite support request, but medium and high altitude satellite support requests can be accommodated by several RTSs because medium and high altitude satellites have more flexible visibility windows. To allow generation of data, we assume that a combination of several RTSs form one pseudo RTS. As a result, we have 33 combinations of RTSs. The 33 pseudo RTSs are listed below.

1 Pohciblgpir : RTSs POGO, HULA, COOK, INDI, BOSS, LION, GUAM, PIKE, and
REEF.

- 2 ilr : RTSs INDI, LION, and REEF.
- 3 ibl : RTSs INDI, BOSS, and LION.
- 4 bl : RTSs BOSS and LION.
- 5 hcbpi : RTSs HULA, COOK, BOSS, and PIKE.
- 6 blpi : RTSs BOSS, LION, and PIKE.
- 7 ilgr : RTSs INDI, LION, GUAM, and REEF.
- 8 pohcblgpi : RTSs POGO, HULA, COOK, BOSS, LION, GUAM, and PIKE.
- 9 cblpi : RTSs COOK, BOSS, LION, and PIKE.
- 10 pohcblgpi : RTSs POGO, HULA, COOK, BOSS, LION, GUAM, and PIKE.
- 11 pohciblgpi : RTSs POGO, HULA, COOK, INDI, BOSS, LION, GUAM, and PIKE.
- 12 igr : RTSs INDI, GUAM, and REEF.
- 13 hcgpi : RTSs HULA, COOK, GUAM, and PIKE.
- 14 hcg : RTSs HULA, COOK, and GUAM.
- 15 pocilgpir : RTSs POGO, HULA, COOK, INDI, LION, GUAM, PIKE, and REEF.
- 16 pocblpi : RTSs POGO, COOK, BOSS, LION, and PIKE.
- 17 pociblp : RTSs POGO, COOK, INDI, BOSS, LION, and PIKE.
- 18 poiblg : RTSs POGO, INDI, BOSS, LION, GUAM, and REEF.
- 19 poiblgpir : RTSs POGO, INDI, BOSS, LION, GUAM, PIKE, and REEF.
- 20 pohcilm : RTSs POGO, HULA, COOK, INDI, LION, PIKE, and REEF.
- 21 pohcbpi : RTSs POGO, HULA, COOK, BOSS, and PIKE.
- 22 hgr : RTSs HULA, GUAM, and REEF.
- 23 cilgpir : RTSs COOK, INDI, LION, GUAM, PIKE, and REEF.
- 24 polgpi : RTSs POGO, LION, GUAM, and PIKE.
- 25 poibr : RTSs POGO, INDI, BOSS, and REEF.
- 26 pohclgpi : RTSs POGO, HULA, COOK, LION, GUAM, and PIKE.
- 27 pohcibgpi : RTSs POGO, HULA, COOK, INDI, BOSS, GUAM, and PIKE.

- 28 pohcg : RTSs POGO, HULA, COOK, and GUAM.
- 29 pohcbgpi : RTSs POGO, HULA, COOK, BOSS, GUAM, and PIKE.
- 30 b : RTS BOSS.
- 31 cibgr : RTSs COOK, INDI, BOSS, GUAM, and REEF.
- 32 cblgpir : RTSs COOK, BOSS, LION, GUAM, PIKE, and REEF.
- 33 g : RTS GUAM.

In Table 3.4, the 33 combinations of RTSs are listed along with the percentage of the medium and high altitude satellite support requests using each combination and the number of requests associated with each combination. The number of requests represents the total for seven days of data.

Table 3.4 The Number of Medium and High Altitude Satellite Support Requests at the Combination of Several RTSs (Total request number is 1148).

combination	pohciblgpir	ilr	ibl	bl	hcbpi	blpi	ilgr	pohcblgpi	cblpi
%	10.10%	8.90%	8.30%	7.70%	7.10%	7.10%	6.50%	4.20%	4.20%
#	116	102	94	88	82	82	73	48	48
combination	pohcblgpi	pohciblgpi	igr	hcgpi	hcg	pocilgpir	pocblpi	pociblpi	poiblgr
%	3.60%	3.60%	3.00%	3.00%	3.00%	2.40%	2.40%	1.80%	1.20%
#	41	41	34	34	34	28	28	21	14
combination	poiblgpir	pohcilpir	pohcbpi	hgr	cilgpir	polgpi	poibr	pohclgpi	pohcibgpi
%	1.20%	1.20%	1.20%	1.20%	1.20%	0.60%	0.60%	0.60%	0.60%
#	14	14	14	14	14	7	7	7	7
combination	pohcg	pohcbgpi	b	cibgr	cblgpir	g			
%	0.60%	0.60%	0.60%	0.60%	0.60%	0.60%			
#	7	7	7	7	7	7			

For medium and high altitude satellite support requests, request length is also complex. Request lengths for medium and high altitude satellite supports range from five minutes to 665 minutes. Table 3.5 shows this.

Table 3.5 The Number of Medium and High Altitude Satellite Supports by Required Length (7 day period).

	D1	D2	D3	D4	D5	D6	D7	TOTAL
5min	23	26	22	25	23	23	22	164
7min	1	0	0	0	0	0	0	1
9min	0	0	1	0	0	0	0	1
10min	39	40	41	41	43	39	38	281
12min	0	0	0	0	0	0	1	1
13min	1	1	0	0	0	0	0	2
14min	0	0	1	0	0	0	0	1
15min	37	34	33	35	35	28	35	237
20min	28	27	29	36	31	27	24	202
23min	0	0	0	0	0	0	1	1
25min	8	6	3	8	3	5	7	40
29min	0	0	0	0	1	0	0	1
30min	1	3	4	2	3	2	4	19
35min	7	5	6	5	5	7	4	39
40min	3	3	4	2	2	5	3	22
45min	11	6	9	12	6	8	11	63
50min	0	1	1	1	2	3	1	9

55min	1	0	0	1	1	0	0	3
57min	1	0	0	0	0	0	0	1
60min	0	0	2	0	1	0	1	4
64min	0	0	0	0	0	1	0	1
74min	0	1	0	0	0	0	0	1
75min	0	1	2	1	0	0	1	5
80min	0	1	0	0	0	1	0	2
85min	0	0	0	0	1	0	0	1
90min	1	1	0	0	0	0	0	2
95min	0	1	1	1	0	0	0	3
100min	1	0	0	0	0	0	0	1
115min	0	0	0	1	0	0	0	1
120min	0	0	1	0	0	0	0	1
145min	0	1	0	0	0	0	0	1
165min	1	1	0	1	1	0	0	4
180min	0	2	1	0	0	0	0	3
185min	0	0	0	0	0	1	0	1
190min	0	0	2	0	0	0	0	2
195min	1	0	0	0	1	0	0	2
255min	0	0	0	0	0	1	0	1
300min	1	1	0	1	1	1	1	6
375min	0	0	1	0	0	0	0	1
405min	0	0	0	0	0	1	0	1
435min	1	1	0	1	1	1	1	6
470min	0	0	0	0	1	0	0	1

475min	0	0	1	0	0	1	0	2
480min	1	1	0	1	0	0	0	3
660min	1	1	0	0	1	0	0	3
665min	0	0	0	1	0	0	0	1
SUM	169	165	165	176	163	155	155	1148

The data shown in Table 3.5 has too large a dispersion to use as a template for the generation of the sample data sets. Thus, we make some assumptions to simplify the problem:

1. A small difference between two close request lengths is ignored.
2. Total number of requests is divided into 14 distinct request lengths.

These assumptions allow establishment of bounds on the number of satellite support requests for a reduced number of request lengths. The results are shown in Table 3.6.

Table 3.6 The Sample Supports Number and Visibility Gap for medium and high altitude satellite.

Req Len (min)	5	10	15	20	25	30	35	40	45	50	165	300	435	660
min #	22	39	30	25	5	2	5	3	12	2	0	0	0	0
max #	26	44	37	35	8	4	7	5	6	4	2	2	2	1
Gap	60	40	35	60	30	10	20	10	5	0	0	0	0	0

In Table 3.6, the first row lists request length, the second row lists the minimum number of requests by required length for the seven day period, and the third row lists the maximum number of requests by required length for the seven day period. As mentioned above, we regrouped the lengths based on the number of requests of each length. The lengths 7, 9, 12, 13, and 14 minutes are only 0.1% of the data sets. We ignore these

lengths. For the same reason, we also ignore lengths of 23, 29, 55, 57, 60, 64, 74, 75, 80, 85, 90, 95, 100, 115, 120, 145, 180, 185, 190, 195, 255, 375, 405, 470, 475, 480, and 665 minutes. Because the medium and high altitude satellite support requests have long visibility windows, we analyzed the available data and determined a value labeled "gap". This value is shown in the last row of Table 3.6. The value gap represents the average difference between support length and the length of the visibility window for the various support lengths. The value gap is used in the following procedure to determine EV, the ending visibility.

Procedure for Generation of the Sample Data for Medium and High Altitude Satellite Support Requests.. The procedure used to generate the sample data sets for medium and high altitude satellite support requests is similar to the approach used in the low altitude satellite case. A flow chart of the procedure is presented in Figure 3.4 and the programs used are presented in Appendix A.

Input Parameters for the Procedure. The input parameters for the procedure are defined as:

I - index for support lengths.

J - index for RTSs.

STz[I] - length of support requests for I=1 to 14.

STn[I] - number of support requests of the particular length for I=1 to 14.

RT[J] - number of support requests at the combination RTS J for J=1 to 33.

TM - total number of the supports requested.

TR - total number of the supports requested at the combination RTSs.

SV - starting time of medium/high altitude satellite visibility window.

EV - ending time of medium/high altitude satellite visibility window.

gap[I] - difference in length between the particular support length STz[I] and the associated visibility window.

Medium and High Altitude Satellite Support Requests

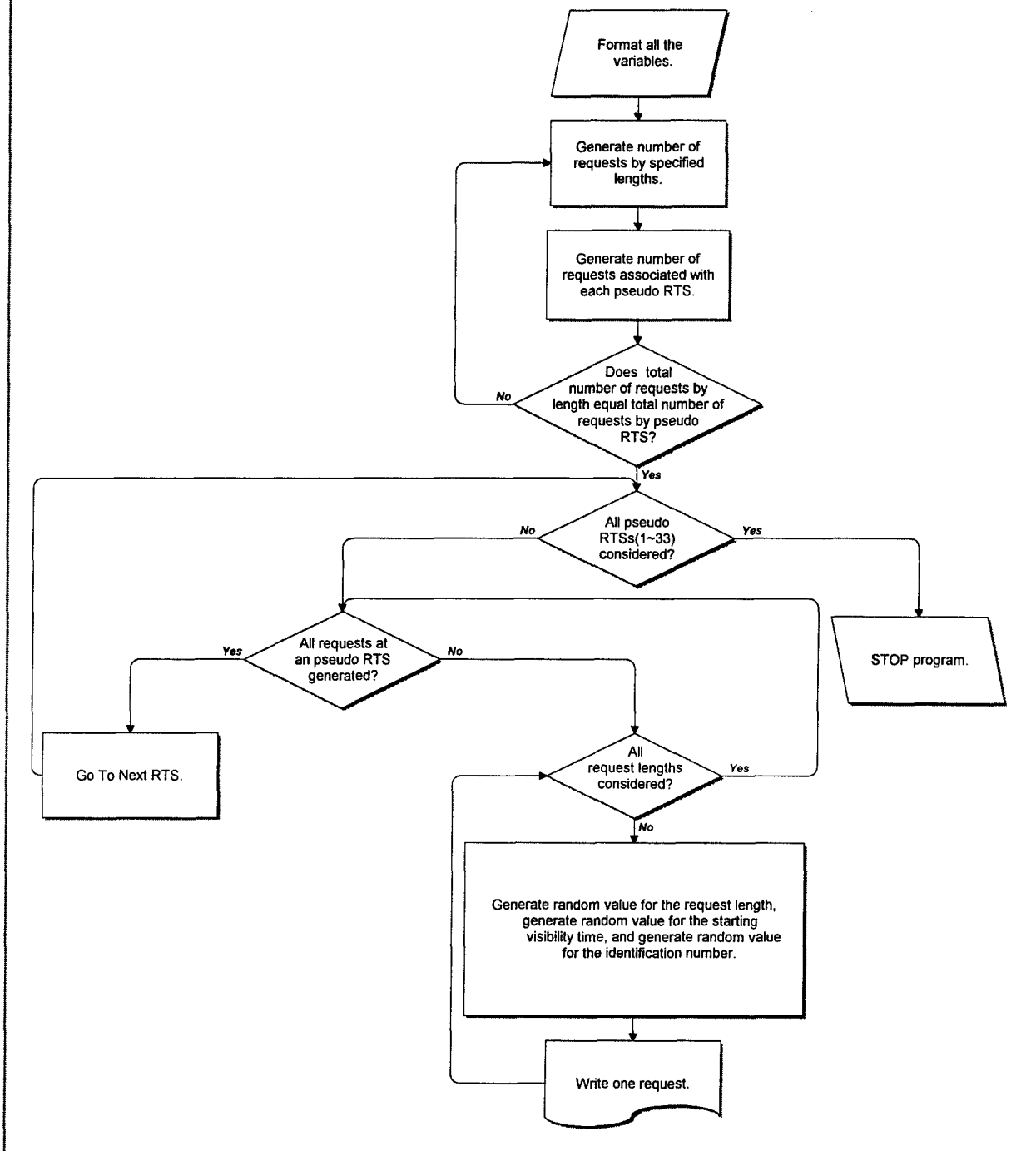


Figure 3.4 Flow Chart for Generating Medium and High Altitude Satellite Support Requests.

Step 0: (Initialization) Put $STz[1] = 5, STz[2] = 10, STz[3] = 15, STz[4] = 20,$
 $STz[5] = 25, STz[6] = 30, STz[7] = 35, STz[8] = 40, STz[9] = 45, STz[10] =$
 $50, STz[11] = 165, STz[12] = 300, STz[13] = 435, STz[14] = 660,$
 $gap[1] = 60, gap[2] = 40, gap[3] = 35, gap[4] = 60, gap[5] = 30, gap[6] = 10,$
 $gap[7] = 20, gap[8] = 10, gap[9] = 5, gap[10] = 0, gap[11] = 0, gap[12] = 0,$
 $gap[13] = 0,$ and $gap[14] = 0.$

Step 1: (Generate random number $STn[I]$)

Set $STn[1] = \text{round}(4 * \text{random} + 22)$ for $22 \leq STn[1] \leq 26,$
 $STn[2] = \text{round}(5 * \text{random} + 39)$ for $39 \leq STn[2] \leq 44,$
 $STn[3] = \text{round}(7 * \text{random} + 30)$ for $30 \leq STn[3] \leq 37,$
 $STn[4] = \text{round}(10 * \text{random} + 25)$ for $25 \leq STn[4] \leq 35,$
 $STn[5] = \text{round}(3 * \text{random} + 5)$ for $5 \leq STn[5] \leq 8,$
 $STn[6] = \text{round}(2 * \text{random} + 2)$ for $2 \leq STn[6] \leq 4,$
 $STn[7] = \text{round}(2 * \text{random} + 5)$ for $5 \leq STn[7] \leq 7,$
 $STn[8] = \text{round}(2 * \text{random} + 3)$ for $3 \leq STn[8] \leq 5,$
 $STn[9] = \text{round}(6 * \text{random} + 6)$ for $6 \leq STn[9] \leq 12,$
 $STn[10] = \text{round}(2 * \text{random} + 2)$ for $2 \leq STn[10] \leq 4,$
 $STn[11] = \text{round}(2 * \text{random})$ for $0 \leq STn[11] \leq 2,$
 $STn[12] = \text{round}(2 * \text{random})$ for $0 \leq STn[12] \leq 2,$
 $STn[13] = \text{round}(2 * \text{random})$ for $0 \leq STn[13] \leq 2,$ and
 $STn[14] = \text{round}(\text{random})$ for $0 \leq STn[14] \leq 1.$

$$TM = \sum_{I=1}^{14} STn[I].$$

Step 2: (Generate number of RT[J]) Set $RT[1] = \text{round}(TM*0.101)$,

$RT[2] = \text{round}(TM*0.089)$, $RT[3] = \text{round}(TM*0.083)$, $RT[4] = \text{round}(TM*0.077)$,
 $RT[5] = \text{round}(TM*0.071)$, $RT[6] = \text{round}(TM*0.071)$, $RT[7] = \text{round}(TM*0.065)$,
 $RT[8] = \text{round}(TM*0.042)$, $RT[9] = \text{round}(TM*0.042)$, $RT[10] = \text{round}(TM*0.036)$,
 $RT[11] = \text{round}(TM*0.036)$, $RT[12] = \text{round}(TM*0.030)$, $RT[13] = \text{round}(TM*0.030)$,
 $RT[14] = \text{round}(TM*0.030)$, $RT[15] = \text{round}(TM*0.024)$, $RT[16] = \text{round}(TM*0.024)$,
 $RT[17] = \text{round}(TM*0.018)$, $RT[18] = \text{round}(TM*0.012)$, $RT[19] = \text{round}(TM*0.012)$,
 $RT[20] = \text{round}(TM*0.012)$, $RT[21] = \text{round}(TM*0.012)$, $RT[22] = \text{round}(TM*0.012)$,
 $RT[23] = \text{round}(TM*0.012)$, $RT[24] = \text{round}(TM*0.006)$, $RT[25] = \text{round}(TM*0.006)$,
 $RT[26] = \text{round}(TM*0.006)$, $RT[27] = \text{round}(TM*0.006)$, $RT[28] = \text{round}(TM*0.006)$,
 $RT[29] = \text{round}(TM*0.006)$, $RT[30] = \text{round}(TM*0.006)$, $RT[31] = \text{round}(TM*0.006)$,
 $RT[32] = \text{round}(TM*0.006)$, and $RT[33] = \text{round}(TM*0.006)$.

$$TR = \sum_{J=1}^{33} RT[J].$$

Step 3: (Backtracking) If $TR \neq TM$, go to Step 1; otherwise, go to Step 4.

Step 4: (Generating RTS) Select $RT[J]$ and J is incremented by one for each repetition
from $J = 1$ to $J = 33$. If $J > 33$, go to Step 9; otherwise, go to Step 5.

Step 5: (Generating number of RTS) K is incremented by one for each repetition from
 $K = 1$ to $K = RT[J]$. If $K > RTn[J]$, go to Step 4; otherwise, go to Step 6.

Step 6: (Generating request length) Randomly select I for $1 \leq I \leq 14$. This gives a particular length $STz[I]$. A Boolean comparison on a counter is done to determine if the number of allowable lengths, $STn[I]$, has been exceeded. If yes, repeat Step 6; otherwise set request length to $STz[I]$ and go to Step 7.

Step 7: (Generating SV and ID) Set $SV = \text{round}(\text{random} * 1440)$ and $ID = \text{round}(\text{random} * 8000 + 1000)$. If generated SV and ID are already used then generate again. Set $EV = SV + STz[I] + \text{gap}[I]$.

Step 8: (Writing the Results) Write the information for the requests to an output file. Then go to Step 5.

Step 9: (Termination) Stop the program.

Step 1 is designed to set the number of supports of length $STn[I]$. This number is generated by a random number function. The number which is used in this step is based on the results from analyzing the ASTRO database. After generating $STn[I]$, add these values and set equal to TM. At Step 2, we generate the number of support requests at the 33 combination RTSs. TR is the sum of the $RT[J]$. At Step 3, if $TR \neq TM$, go to Step 1 and reaccomplish the generation. This step avoids the problem of the number of support requests not equaling the number of requests allocated to the RTSs. At Step 4, $RT[J]$ is set to the number of support requests at the Jth pseudo RTS. There are 33 pseudo RTSs. J is incremented by one for each repetition of this Step until $J = 33$. At Step 5, each pseudo RTS has a fixed number of satellite support requests. K is incremented by one for each repetition of this step until K reaches the total number of satellite support requests at the designated pseudo RTS. At Step 6, the value I is generated randomly and used to

designate a particular support length $ST_z[I]$. A check is also done to insure that $ST_n[I]$, the number of support of a particular length, has not been exceeded. At Step 7, SV and ID are generated by using the random number function. At this step, EV is set equal to the starting visibility, SV, plus the support length generated at Step 6 plus the appropriate gap. At Step 8, the information from the previous steps needed to describe a satellite support request is written to a file. At Step 9, if all conditions in Step 4 are satisfied, the program stops. A sample data set is shown in Appendix D.

Schedule and Analyze The Sample Data Sets Using Schalck's SRS Scheduling Algorithm. To determine the capacity of the AFSCN, we will use Schalck's scheduling algorithm to find the number of supports scheduled for various size data sets. This will be accomplished in the following manner and is shown in Figure 3.5.

1. Determine the number of low and medium/high altitude satellite support requests.
2. Create a sample data set for low altitude satellite support requests.
3. Create a sample data set for medium and high altitude satellite support requests.
4. Apply Schalck's SRS scheduling algorithm to the sample data and generate schedules.
5. Analyze the results and determine the capacity of the AFSCN.

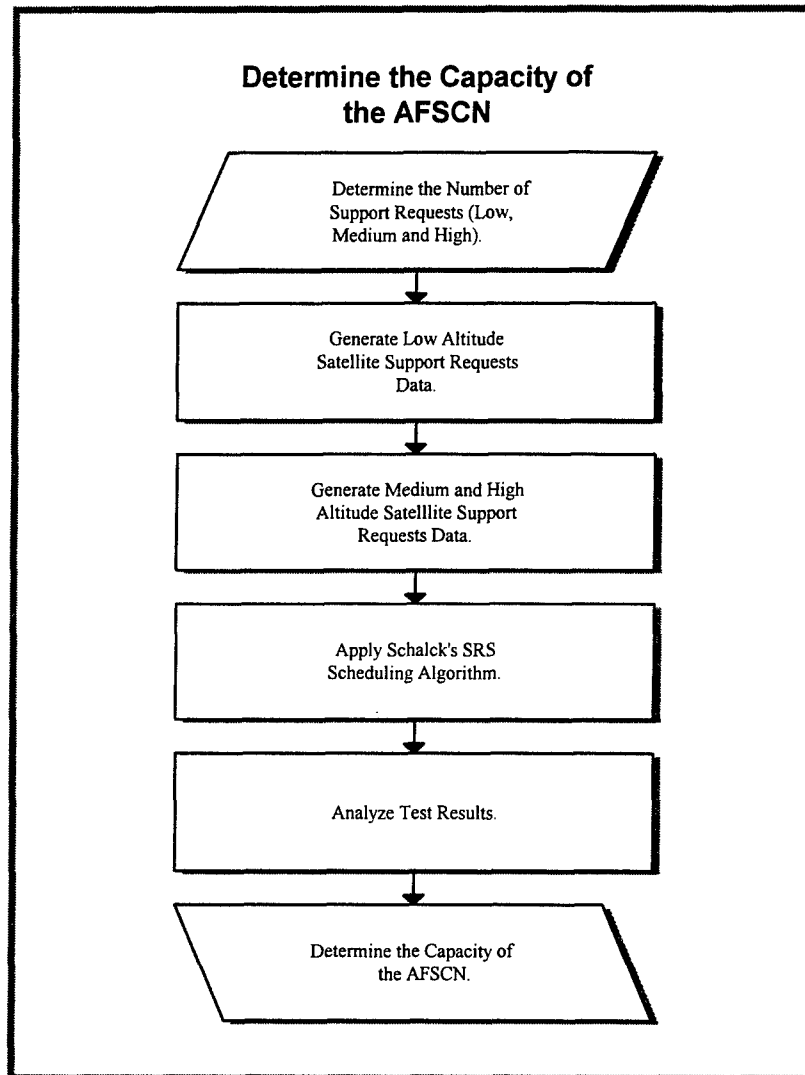


Figure 3.5 Determine the Capacity of AFSCN.

IV. RESULTS

Overview

This chapter details the results of tests of the procedure for determining the capacity of the AFSCN described in Chapter 3. First, the generation of low altitude and medium/high altitude satellite support requests sample data set generation are discussed, followed by a discussion of the scheduling performance for each case. Scheduling performance is a measure of the number of supports requested and the number of supports scheduled. Finally, the results of scheduling the sample data sets are presented.

Sample Data Sets Generation

This section details the results of sample data set generation. The sample data sets are based on actual AFSCN data for a one-week period. The data covers the week of 12 October 1992 to 18 October 1992 and is the same data Schalck used in his research (3). The sets of data were provided by Ken Chambers, technical advisor, 21 SOPS, Onizuka AFB, CA.

As mentioned in Chapter 3, two procedures for generation of the support requests sample data sets were used. One generates the low altitude satellite support requests sample data sets and the other generates the medium and high altitude satellite support requests sample data sets.

Results of generation of the low altitude satellite support requests sample data sets. The results of the generation procedure were very similar to the databases which were used in Schalck's research. Table 4.1 provides sample low altitude support entries in the database.

Table 4.1 Sample Data Sets for Low Altitude Satellite Support.

Sup No.	RTS	BV	EV	Req Len	TAT	Iden & Rev
1	BOSS-A	101	151	50	20	747701111.1
1	BOSS-B	101	151	50	20	747701111.1
2	BOSS-A	565	577	12	20	780801111.1
2	BOSS-B	565	577	12	20	780801111.1
3	BOSS-A	316	321	5	20	757801111.1
3	BOSS-B	316	321	5	20	757801111.1
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
144	REEF-A	1156	1171	15	20	837501111.1
145	REEF-A	1381	1397	16	20	200101111.1

The first column shows the support number, the second column is the RTS, the third and fourth columns show the beginning visibility time and ending visibility time, the fifth column is the turn-around or set-up time, and the sixth column presents a combination of the identification number and revolution number. We assumed the revolution number was fixed at 1111.1. In the low altitude satellite case, turn-around time is fixed at 20 minutes.

To test the capacity of the AFSCN, we increased the number of satellite support requests by 17.5% and then by 35% from the normal number of requests. For each level of increase, we generated 15 sample data sets. Because of the random fashion in which

the sample data sets were generated, the increases from the normal level do not exactly match the specified percentage increases. The results of sample data set generation are shown in Table 4.2.

Table 4.2 The Number of Daily Low Altitude Satellite Support Requests.

test #	Normal	Up 17.5%	Up 35%
1	145	172	197
2	143	184	188
3	150	162	209
4	145	174	190
5	153	167	216
6	131	175	181
7	144	160	189
8	147	174	201
9	150	169	198
10	143	173	205
11	146	177	202
12	149	175	191
13	150	173	207
14	148	167	211
15	151	180	203

Results of generation of the medium/high altitude satellite support requests sample data sets. The medium and high altitude satellite support requests sample data set generation has the same format as the one used in low altitude satellite support requests

case. Table 4.3 shows the format of the medium and high altitude satellite support requests sample data sets.

Table 4.3 Sample Data Sets for Medium and High Altitude Satellite Support.

Sup No.	RTS	BV	EV	Req Len	TAT	Iden & Rev
1	POGO-A	1295	1345	15	15	706900000.0
1	POGO-B	1295	1345	15	15	706900000.0
1	POGO-C	1295	1345	15	15	706900000.0
1	HULA-A	1295	1345	15	15	706900000.0
1	HULA-B	1295	1345	15	15	706900000.0
1	COOK-A	1295	1345	15	15	706900000.0
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
167	PIKE-A	1380	1430	10	15	181600000.0
167	REEF-A	1380	1430	10	15	181600000.0

For the medium and high altitude satellite support requests, we fixed turn-around time at 15 minutes. We also assumed the revolution number was fixed at 0000.0.

We generated the sample data sets using the procedure outlined in Chapter 3. The number of support requests generated is increased by 35% and 50% from the normal level. We also generated 15 data sets for each level. The number of support requests generated are shown in Table 4.4.

Table 4.4 The Number of Daily Medium/High Altitude Satellite Support Requests.

test #	Normal	Up 35%	Up 50%
1	167	218	246
2	159	212	246
3	160	219	248
4	167	211	249
5	167	215	246
6	166	218	250
7	160	216	247
8	167	217	246
9	166	213	245
10	161	218	249
11	167	212	247
12	166	216	250
13	167	215	251
14	168	214	248
15	168	213	245

Sorting the Data Sets. The data sets from the generation procedure cannot be used directly in the scheduling algorithm because we used random numbers to generate the beginning visibility time. Therefore, the beginning visibility times are not generated in an ascending order. To use the scheduling algorithm, we have to sort the beginning visibility time in ascending order. A detailed program code and explanation of the sorting are provided in Appendix B.

Scheduling Results. There are several ways to schedule the satellite support requests and we selected Schalck's approach (3). The programs required to schedule satellite support requests using Schalck's approach are shown in Appendix C. We selected Schalck's algorithm because it is more effective than Gooley's approach, just as effective as Parish's approach and easy to use. Turbo Pascal For Windows was chosen to generate the sample data sets because of its random number generation, debugging features, and its compatibility with the available data sets. In the following, scheduling performance is the number of satellite support requests scheduled divided by the number of satellite support requests. An example of a schedule is presented in Appendix E.

Nine combinations or cases of the low and medium/high altitude satellite support requests were tested. A case consists of a specified level of low altitude satellite support requests and a specified level of medium/high altitude satellite support requests. The cases are defined below, unless otherwise stated, the normal number of requests is used. For the low altitude satellites, the normal number of requests is approximately 144. For the medium/high altitude satellites, the normal number of requests is approximately 164. Because of the random nature of data set generation, variations from these values were observed.

Case 1) the normal number of low altitude satellite support requests and the normal number of medium/high altitude satellite support requests.

Case 2) increase the number of low altitude satellite support requests by 17.5% from the normal number of low altitude satellite support requests.

Case 3) increase the number of low altitude satellite support requests by 35% from the normal number of low altitude satellite support requests.

Case 4) increase the number of medium/high altitude satellite support requests by 35% from the normal number of medium/high altitude satellite support requests.

Case 5) increase the number of low altitude satellite support requests by 17.5% from the normal number of low altitude satellite support requests and increase the number of medium/high altitude satellite support requests by 35% from the normal number of medium/high altitude satellite support requests.

Case 6) increase the number of low altitude satellite support requests by 35% from the normal number of low altitude satellite support requests and increase the number of medium/high altitude satellite support requests by 35% from the normal number of medium/high altitude satellite support requests.

Case 7) increase the number of medium/high altitude satellite support requests by 50% from the normal number of medium/high altitude satellite support requests.

Case 8) increase the number of low altitude satellite support requests by 17.5% from the normal number of low altitude satellite support requests and increase the number of medium/high altitude satellite support requests by 50% from the normal number of medium/high altitude satellite support requests.

Case 9) increase the number of low altitude satellite support requests by 35% from the normal number of low altitude satellite support requests and increase the number of medium/high altitude satellite support requests by 50% from the normal number of medium/high altitude satellite support requests.

Tables 4.5 - 4.25 show the scheduling results for the cases.

**Table 4.5 Scheduling Performance for the Low Altitude Satellite Support
Requests: Case 1.**

test #	request #	schedule #	schedule %
1	145	135	93.103%
2	143	136	95.105%
3	150	139	92.667%
4	145	138	95.172%
5	153	141	92.157%
6	131	125	95.420%
7	144	132	91.667%
8	147	138	93.878%
9	150	139	92.667%
10	143	134	93.706%
11	146	135	92.466%
12	149	137	91.946%
13	150	140	93.333%
14	148	135	91.216%
15	151	139	92.053%

**Table 4.6 Scheduling Performance for the Medium/High Altitude Satellite Support
Requests: Case 1.**

test #	request #	schedule #	schedule %
1	167	165	98.802%
2	159	155	97.484%
3	160	155	96.875%
4	167	161	96.407%
5	167	162	97.006%
6	166	164	98.795%
7	160	158	98.750%
8	167	162	97.006%
9	166	163	98.193%
10	161	160	99.379%
11	167	164	98.204%
12	166	160	96.386%
13	167	163	97.605%
14	168	161	95.833%
15	166	164	98.795%

**Table 4.7 Scheduling Performance for All Satellite Support
Requests: Case 1.**

test #	request #	schedule #	schedule %
1	312	300	96.154%
2	302	291	96.358%
3	310	294	94.839%
4	312	299	95.833%
5	320	303	94.688%
6	297	289	97.306%
7	304	290	95.395%
8	314	300	95.541%
9	316	302	95.570%
10	304	294	96.711%
11	313	299	95.527%
12	315	297	94.286%
13	317	303	95.584%
14	316	296	93.671%
15	317	303	95.584%

**Table 4.8 Scheduling Performance for the Low Altitude Satellite Support
Requests: Case 2.**

test #	request #	schedule #	schedule %
1	172	156	90.698%
2	184	164	89.130%
3	162	147	90.741%
4	174	157	90.230%
5	167	153	91.617%
6	175	158	90.286%
7	160	150	93.750%
8	174	159	91.379%
9	169	153	90.533%
10	173	156	90.173%
11	177	160	90.395%
12	175	159	90.857%
13	173	156	90.173%
14	167	152	91.018%
15	180	163	90.556%

**Table 4.9 Scheduling Performance for the Medium/High Altitude Satellite Support
Requests: Case 2.**

test #	request #	schedule #	schedule %
1	167	161	96.407%
2	159	151	94.969%
3	160	151	94.375%
4	167	164	98.204%
5	167	161	96.407%
6	166	161	96.988%
7	160	155	96.875%
8	167	162	97.006%
9	166	161	96.988%
10	161	153	95.031%
11	167	159	95.210%
12	166	161	96.988%
13	167	160	95.808%
14	168	161	95.833%
15	166	159	95.783%

**Table 4.10 Scheduling Performance for All Satellite Support
Requests: Case 2.**

test #	request #	schedule #	schedule %
1	339	317	93.510%
2	343	315	91.837%
3	322	298	92.547%
4	341	321	94.135%
5	334	314	94.012%
6	341	319	93.548%
7	320	305	95.313%
8	341	321	94.135%
9	335	314	93.731%
10	334	309	92.515%
11	344	319	92.733%
12	341	320	93.842%
13	340	316	92.941%
14	335	313	93.433%
15	346	322	93.064%

**Table 4.11 Scheduling Performance for the Low Altitude Satellite Support
Requests: Case 3.**

test #	request #	schedule #	schedule %
1	197	172	87.310%
2	188	167	88.830%
3	209	179	85.646%
4	190	169	88.947%
5	216	184	85.185%
6	181	163	90.055%
7	189	169	89.418%
8	201	176	87.562%
9	198	171	86.364%
10	205	179	87.317%
11	202	175	86.634%
12	191	170	89.005%
13	207	178	85.990%
14	211	181	85.782%
15	203	177	87.192%

**Table 4.12 Scheduling Performance for the Medium/High Altitude Satellite Support
Requests: Case 3.**

test #	request #	schedule #	schedule %
1	167	164	98.204%
2	159	158	99.371%
3	160	151	94.375%
4	167	164	98.204%
5	167	161	96.407%
6	166	161	96.988%
7	160	151	94.375%
8	167	162	97.006%
9	166	163	98.193%
10	161	157	97.516%
11	167	163	97.605%
12	166	164	98.795%
13	167	164	98.204%
14	168	162	96.429%
15	166	163	98.193%

**Table 4.13 Scheduling Performance for All Satellite Support
Requests: Case 3.**

test #	request #	schedule #	schedule %
1	364	336	92.308%
2	347	325	93.660%
3	369	330	89.431%
4	357	333	93.277%
5	383	345	90.078%
6	347	324	93.372%
7	349	320	91.691%
8	368	338	91.848%
9	364	334	91.758%
10	366	336	91.803%
11	369	338	91.599%
12	357	334	93.557%
13	374	342	91.444%
14	379	343	90.501%
15	369	340	92.141%

Scheduling performance for the low altitude satellite support requests for Case 4 is the same as Case 1.

Table 4.14 Scheduling Performance for the Medium/High Altitude Satellite Support Requests: Case 4.

test #	request #	schedule #	schedule %
1	218	207	94.954%
2	212	201	94.811%
3	219	204	93.151%
4	211	197	93.365%
5	215	210	97.674%
6	218	207	94.954%
7	216	197	91.204%
8	217	202	93.088%
9	213	201	94.366%
10	218	199	91.284%
11	212	203	95.755%
12	216	207	95.833%
13	215	201	93.488%
14	214	203	94.860%
15	213	202	94.836%

**Table 4.15 Scheduling Performance for All Satellite Support
Requests: Case 4.**

test #	request #	schedule #	schedule %
1	363	342	94.215%
2	355	337	94.930%
3	369	343	92.954%
4	356	335	94.101%
5	368	351	95.380%
6	349	332	95.129%
7	360	329	91.389%
8	364	340	93.407%
9	363	340	93.664%
10	361	333	92.244%
11	358	338	94.413%
12	365	344	94.247%
13	365	341	93.425%
14	362	338	93.370%
15	364	341	93.681%

Scheduling performance for the low altitude satellite support requests for Case 5 is the same as Case 2.

Table 4.16 Scheduling Performance for the Medium/High Altitude Satellite Support Requests: Case 5.

test #	request #	schedule #	schedule %
1	218	201	92.202%
2	212	194	91.509%
3	219	204	93.151%
4	211	197	93.365%
5	215	204	94.884%
6	218	201	92.202%
7	216	201	93.056%
8	217	202	93.088%
9	213	198	92.958%
10	218	203	93.119%
11	212	196	92.453%
12	216	201	93.056%
13	215	196	91.163%
14	214	197	92.056%
15	213	199	93.427%

**Table 4.17 Scheduling Performance for All Satellite Support
Requests: Case 5.**

test #	request #	schedule #	schedule %
1	390	357	91.538%
2	396	358	90.404%
3	381	351	92.126%
4	385	354	91.948%
5	382	357	93.455%
6	393	359	91.349%
7	376	351	93.351%
8	391	361	92.327%
9	382	351	91.885%
10	391	359	91.816%
11	389	356	91.517%
12	391	360	92.072%
13	388	352	90.722%
14	381	349	91.601%
15	393	362	92.112%

Scheduling Performance for the low altitude satellite support requests for Case 6 is the same as Case 3.

Table 4.18 Scheduling Performance for the Medium/High Altitude Satellite Support Requests: Case 6.

test #	request #	schedule #	schedule %
1	218	204	93.578%
2	212	197	92.925%
3	219	197	89.954%
4	211	197	93.365%
5	215	197	91.628%
6	218	194	88.991%
7	216	201	93.056%
8	217	200	92.166%
9	213	195	91.549%
10	218	199	91.284%
11	212	197	92.925%
12	216	195	90.278%
13	215	193	89.767%
14	214	191	89.252%
15	213	194	91.080%

**Table 4.19 Scheduling Performance for All Satellite Support
Requests: Case 6.**

test #	request #	schedule #	schedule %
1	415	376	90.602%
2	400	364	91.000%
3	428	376	87.850%
4	401	366	91.272%
5	431	381	88.399%
6	399	357	89.474%
7	405	370	91.358%
8	418	376	89.952%
9	411	366	89.051%
10	423	378	89.362%
11	414	372	89.855%
12	407	365	89.681%
13	422	371	87.915%
14	425	372	87.529%
15	416	371	89.183%

Scheduling performance for the low altitude satellite support requests for Case 7 is the same as Case 1.

Table 4.20 Scheduling Performance for the Medium/High Altitude Satellite Support Requests: Case 7.

test #	request #	schedule #	schedule %
1	246	234	95.122%
2	246	231	93.902%
3	248	232	93.548%
4	249	233	93.574%
5	246	230	93.496%
6	250	236	94.400%
7	247	229	92.713%
8	246	231	93.902%
9	245	230	93.878%
10	249	232	93.173%
11	247	233	94.332%
12	250	229	91.600%
13	251	232	92.430%
14	248	236	95.161%
15	245	229	93.469%

**Table 4.21 Scheduling Performance for All Satellite Support
Requests: Case 7.**

test #	request #	schedule #	schedule %
1	391	369	94.373%
2	389	367	94.344%
3	398	371	93.216%
4	394	371	94.162%
5	399	371	92.982%
6	381	361	94.751%
7	391	361	92.327%
8	393	369	93.893%
9	395	369	93.418%
10	392	366	93.367%
11	393	368	93.639%
12	399	366	91.729%
13	401	372	92.768%
14	396	371	93.687%
15	396	368	92.929%

Scheduling performance for the low altitude satellite support requests for Case 8 is the same as Case 2.

Table 4.22 Scheduling Performance for the Medium/High Altitude Satellite Support Requests: Case 8.

test #	request #	schedule #	schedule %
1	246	228	92.683%
2	246	224	91.057%
3	248	225	90.726%
4	249	230	92.369%
5	246	226	91.870%
6	250	226	90.400%
7	247	228	92.308%
8	246	225	91.463%
9	245	227	92.653%
10	249	226	90.763%
11	247	228	92.308%
12	250	229	91.600%
13	251	228	90.837%
14	248	228	91.935%
15	245	225	91.837%

**Table 4.23 Scheduling Performance for All Satellite Support
Requests: Case 8.**

test #	request #	schedule #	schedule %
1	418	384	91.866%
2	430	388	90.233%
3	410	372	90.732%
4	423	387	91.489%
5	413	379	91.768%
6	425	384	90.353%
7	407	378	92.875%
8	420	384	91.429%
9	414	380	91.787%
10	422	382	90.521%
11	424	388	91.509%
12	425	388	91.294%
13	424	384	90.566%
14	415	380	91.566%
15	425	388	91.294%

Scheduling performance for the low altitude satellite support requests for Case 9 is the same as Case 3.

Table 4.24 Scheduling Performance for the Medium/High Altitude Satellite Support Requests: Case 9.

test #	request #	schedule #	schedule %
1	246	226	91.870%
2	246	227	92.276%
3	248	228	91.935%
4	249	230	92.369%
5	246	228	92.683%
6	250	225	90.000%
7	247	228	92.308%
8	246	229	93.089%
9	245	226	92.245%
10	249	228	91.566%
11	247	227	91.903%
12	250	230	92.000%
13	251	228	90.837%
14	248	226	91.129%
15	245	228	93.061%

**Table 4.25 Scheduling Performance for All Satellite Support
Requests: Case 9.**

test #	request #	schedule #	schedule %
1	443	398	89.842%
2	434	394	90.783%
3	457	407	89.059%
4	439	399	90.888%
5	462	412	89.177%
6	431	388	90.023%
7	436	397	91.055%
8	447	405	90.604%
9	443	397	89.616%
10	454	407	89.648%
11	449	402	89.532%
12	441	400	90.703%
13	458	406	88.646%
14	459	407	88.671%
15	448	405	90.402%

Summary of Results. Figure 4.1 shows the scheduling performance for low altitude satellite support requests. As shown before in the Tables 4.5, 4.6, and 4.7, the number of low altitude satellite support requests range from 131 to 216. The percentage of low altitude satellite support requests scheduled over this range is plotted in Figure 4.1. From the plot, it is seen that the percentage scheduled decreases as the number of low altitude satellite support requests increases. We also accomplished a least squares regression. Using the regression equation, we found that 90 percent scheduling performance was achieved for 176.6 low altitude satellite support requests. The regression line and 95 percent confidence interval are also shown in Figure 4.1.

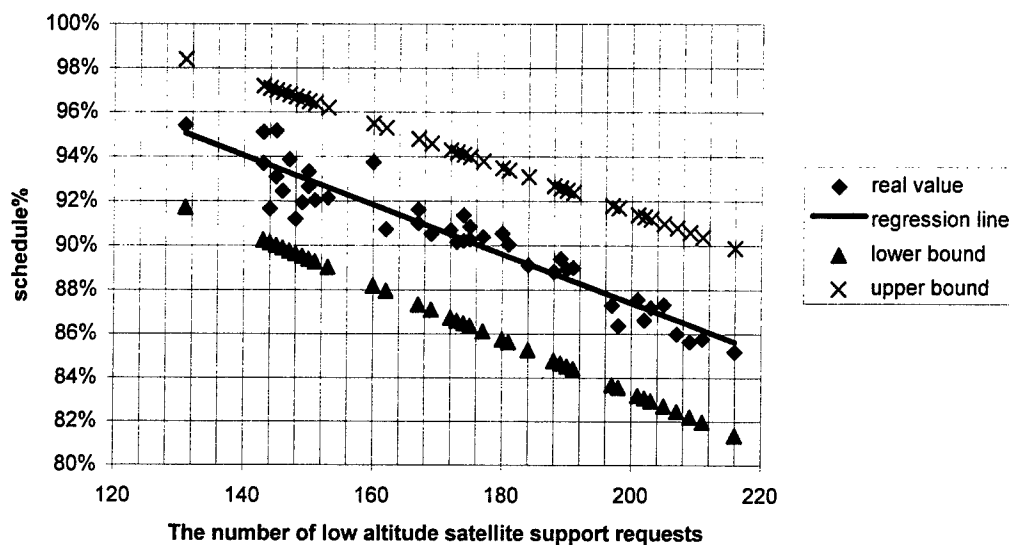


Figure 4.1 The Scheduling Performance for Low Altitude Satellite Support Requests.

In the medium and high altitude satellite case, the scheduling performance depends strongly on the scheduling performance for the low altitude satellite support requests, because the low altitude satellite support requests were scheduled before attempting to schedule the medium and high altitude satellite support requests.

Figures 4.2, 4.3, and 4.4 show the scheduling performance for medium and high altitude satellite support requests as the number of low altitude satellite support requests increase. The scheduling performance for medium and high altitude satellite support requests is above 90% in each case. Figure 4.2 shows the scheduling performance for the medium/high altitude satellite support requests when there are about 145 low altitude satellite support requests. The least squares regression line and the 95 percent confidence interval are also shown. The upper confidence interval is bounded at 100 percent scheduling performance. Figure 4.3 shows the scheduling for the medium/high altitude satellite support requests when there are about 172 low altitude satellite support requests. Figure 4.3 shows the scheduling for the medium/high altitude satellite support requests when there are about 199 low altitude satellite support requests.

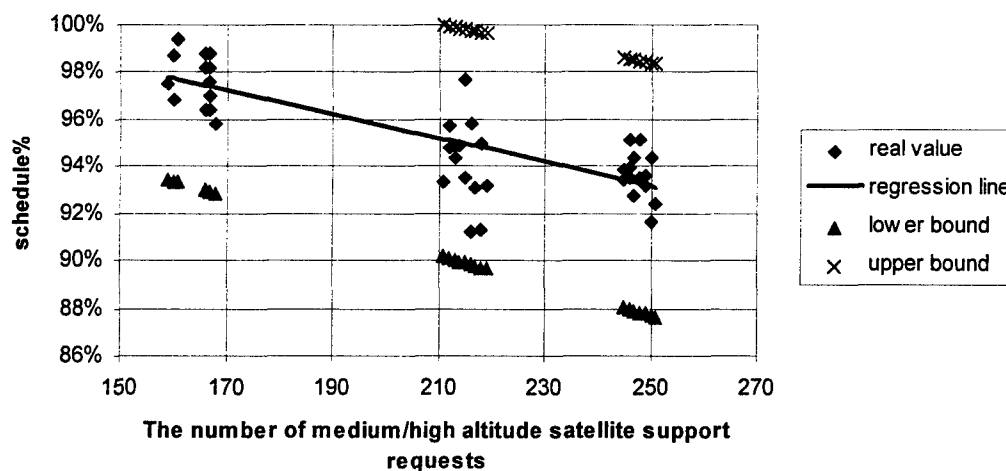


Figure 4.2 The Scheduling Performance for Medium and High Altitude Satellite Support Requests when the Number of Low Altitude Satellite Support Requests is around 145.

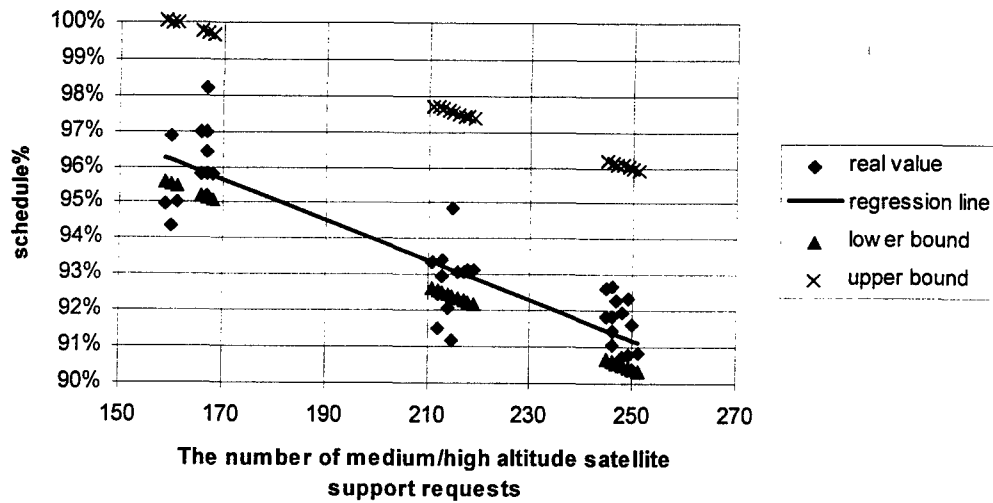


Figure 4.3 The Scheduling Performance for Medium and High Altitude Satellite Support Requests when the Number of Low Altitude Satellite Support Requests is increased by 17.5%.

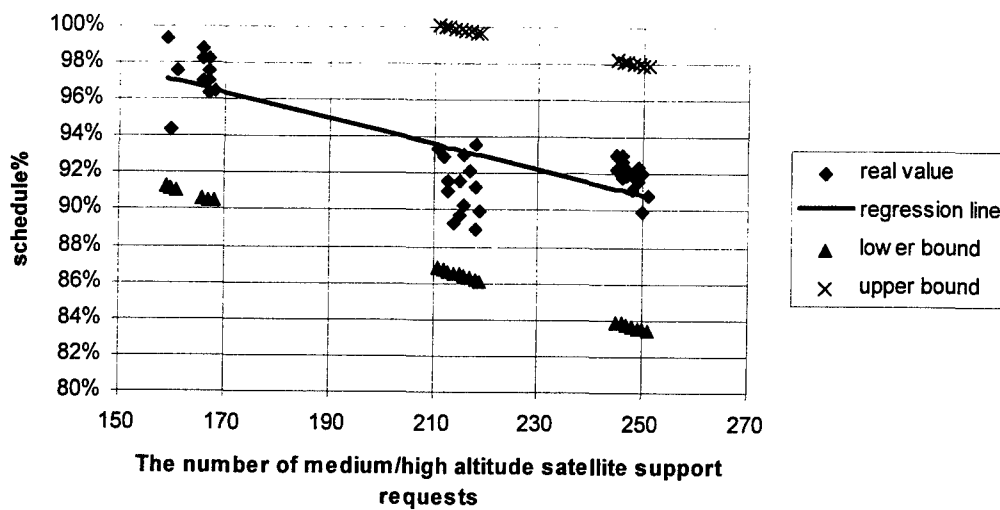


Figure 4.4 The Scheduling Performance for Medium and High Altitude Satellite Support Requests when the Number of Low Altitude Satellite Support Requests is increased by 35%.

When the number of low altitude satellite support requests was increased by 17.5 percent from the normal level, we averaged 172.1 requests in our sample data sets. At this level the scheduling performance for all satellite support requests was about 90 percent. Looking at the medium and high altitude satellite support requests, with the number of low altitude satellite support requests fixed at a 17.5 percent increase or a 35 percent increase, we achieved the results shown in Table 4.26.

Table 4.26 Average Scheduling Performance for All Requests.

	Normal	Up 35%	Up 50%
Ave # med/hi	165.07	215.13	247.53
Ave Sch Perf (17.5%)	93.42	91.88	91.22
Ave Sch Perf (35%)	91.90	89.50	89.93

Table 4.26 shows the average number of medium/high altitude satellite support requests for the normal level and increases of 35 and 50 percent in the number of medium/high altitude satellite support requests. In Table 4.26, we also show the average scheduling performance for all satellite support requests given the 17.5 and 35 percent increases in low altitude satellite support requests from the normal level and the indicated levels for medium/high altitude satellite support requests. As can be seen in Table 4.26, medium/high altitude satellite support requests scheduling performance depends upon the number of low altitude satellite support requests. When low altitude satellite support requests are increased from the normal level by 35 percent, 90 percent scheduling performance cannot be achieved for medium/high altitude satellite support requests above the normal level. Since the average number of low altitude satellite support requests for the 17.5 percent level, 172.1, is close to the 90 percent scheduling

performance level from the regression, 176.6, we infer from Table 4.26 that we can schedule about 175 low altitude satellite support requests and 250 medium/high altitude satellite support requests with approximately 90 percent scheduling performance for all satellite support requests.

V. CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The motivation for this research was to determine the capacity of the AFSCN using an automated algorithm to schedule the satellite support requests. As mentioned before, we used Schalck's scheduling algorithm to schedule the requests in the sample data sets. Before scheduling, we generate the sample data sets. We successfully generated the sample data sets for both low altitude and medium/high altitude satellite support requests and then used Schalck's algorithm to accomplish the scheduling. Schalck mentioned the limitation of his algorithm in his research (3:4-14). Those limitations also apply to this research.

The limitations are:

- RTS down-times and long DPAD supports were not included in the schedules.
- Low altitude satellite turn-around time is fixed at 20 minutes.
- Medium and high altitude satellites turn-around time is fixed at 15 minutes.

Low altitude satellite support requests usually take precedence over the medium and high altitude satellite support requests. We assumed the acceptable scheduling performance is 90%, so the bound on the number of low altitude satellite support requests scheduled at 90% is around 175. The scheduling performance for medium and high altitude satellite support requests is more complicated. Looking at Figures 4.2, 4.3, and 4.4, the results show that the scheduling performance for medium and high altitude satellite support requests remains above 90%. Schalck's algorithm is not able to schedule the medium and high altitude satellite support requests when the number of medium and high altitude satellite support requests exceeds 260. However, the results show the scheduling performance for all satellite support requests is about 90% when there are 175

low altitude satellite support requests and 250 medium and high altitude satellite support requests.

Recommendations

A procedure needs to be developed for scheduling requests that include the RTS downtimes and long term requests. This allows more accurate analysis. Also, a procedure needs to be created that determines RTS utilization.

APPENDIX A

This appendix details the satellite support request sample data sets generation programs used to generate a single day's sample satellite support requests.

The sample data sets generation procedures were programmed in Borland Turbo Pascal For Windows. Turbo Pascal For Windows was run on a PC 486DX66 computer.

Low altitude satellite support requests sample data sets generation code.

This code was developed to generate low altitude satellite support request sample data sets. The code is shown below:

```
Program lreqrn;  
  Uses winCrt;  
  Var  
  
    STz :array[1..14] of integer;  
    STn :array[1..14] of integer;  
    RT : array[1..9] of String[5];  
    RTn : array[1..9] of integer;  
    SV : array[1..1000] of longint;  
    ID : array[1..1000] of longint;  
    nstz:array[1..14] of integer;  
    I,J,k:integer;  
    TM,TR:integer;  
    sn:integer;  
    r,EV:integer;  
    tmp:integer;  
    used,true,false,isv,iid:integer;  
  
  procedure writedat;  
  begin  
    If I=8 then  
      Begin  
        writeln(Output,sn:4,' ',RT[I],'A',SV[sn]:5,EV:5,R:5,' 20 ',ID[sn],'01111.1');  
        writeln(Output,sn:4,' ',RT[I],'B',SV[sn]:5,EV:5,R:5,' 20 ',ID[sn],'01111.1');
```

```

writeln(Output,sn:4,' ',RT[I],'C',SV[sn]:5,EV:5,R:5,' 20 ',ID[sn],'01111.1');
sn:=sn+1
end;

If (I=5) or (I=7) or (I=9) then
Begin
writeln(Output,sn:4,' ',RT[I],'A',SV[sn]:5,EV:5,R:5,' 20 ',ID[sn],'01111.1');
sn:=sn+1
end;

If (I=1) or (I=2) or (I=3) or (I=4) or (I=6) then
Begin
writeln(Output,sn:4,' ',RT[I],'A',SV[sn]:5,EV:5,R:5,' 20 ',ID[sn],'01111.1');
writeln(Output,sn:4,' ',RT[I],'B',SV[sn]:5,EV:5,R:5,' 20 ',ID[sn],'01111.1');
sn:=sn+1
end;
end;

Begin
Assign(output,'A:\lr.dat');
Rewrite(Output);
STz[1]:=5; STz[2]:=9; STz[3]:=10; STz[4]:=11; STz[5]:=12; STz[6]:=13;
STz[7]:=14; STz[8]:=15; STz[9]:=16; STz[10]:=17; STz[11]:=18; STz[12]:=40;
STz[13]:=45; STz[14]:=50;
RT[1]:='BOSS-'; RT[2]:='COOK-'; RT[3]:='GUAM-'; RT[4]:='HULA-'; RT[5]:='INDI-';
RT[6]:='LION-'; RT[7]:='PIKE-'; RT[8]:='POGO-'; RT[9]:='REEF-';

true:=1;
false:=0;

repeat

Randomize;

STn[1] := round(random);      {from 0 to 1}
STn[2] := round(2*random);    {from 0 to 2}
STn[3] := round(4*random+2);  {from 2 to 6}
STn[4] := round(6*random+5);  {from 5 to 11}
STn[5] := round(9*random+7);  {from 7 to 16}
STn[6] := round(11*random+10); {from 10 to 21}
STn[7] := round(20*random+19); {from 19 to 39}
STn[8] := round(11*random+22); {from 22 to 33}
STn[9] := round(15*random+28); {from 28 to 43}
STn[10] := round(9*random+6);  {from 6 to 15}
STn[11] := round(random);      {from 0 to 1}
STn[12] := round(random);      {from 0 to 1}
STn[13] := round(2*random);    {from 0 to 2}
STn[14] := round(random);      {from 0 to 1}

TM:=0;
for I:=1 to 14 do
begin
TM:=TM+STn[I];
end;

```

```

RTn[1] := round(7*random+16); {from 16 to 23}
RTn[2] := round(4*random+8); {from 8 to 12}
RTn[3] := round(6*random+8); {from 8 to 14}
RTn[4] := round(7*random+13); {from 13 to 20}
RTn[5] := round(4*random+3); {from 3 to 7}
RTn[6] := round(10*random+14); {from 14 to 24}
RTn[7] := round(6*random+6); {from 6 to 12}
RTn[8] := round(21*random+36); {from 36 to 57}
RTn[9] := round(5*random+8); {from 8 to 13}

```

```

TR:=0;
for I:=1 to 9 do
begin
  TR:=TR+RTn[I];
end;

```

```

until TM=TR;

```

```

for I:=1 to 14 do
begin
  nstz[I]:=0;
end;

```

```

sn:=1;

```

```

for I:=1 to 9 do
begin

```

```

  for J:=1 to RTn[I] do

```

```

    begin

```

```

      used:=false;

```

```

      while (used=false) do

```

```

        begin

```

```

          tmp:=round(13*random+1);

```

```

          if (nstz[tmp]<STn[tmp]) then

```

```

            used:=true;

```

```

          end;

```

```

        R:=stz[tmp];

```

```

        nstz[tmp]:=nstz[tmp]+1;

```

```

      used := true;

```

```

      while (used = true) do

```

```

        begin

```

```

          isv:=round(1440*random);

```

```

          EV := isv + R;

```

```

          used := false;

```

```

        for k:= 1 to sn do

```

```

          IF (isv = SV[k]) or (EV>1440) then

```

```

        used := true;

    end;
    SV[sn] := isv;

    used := true;
    while (used = true) do
    begin
        iid:=round(8000*random+1000);

        used := false;
        for k:= 1 to sn do
            IF (iid = ID[k]) then
                used := true;
            end;
        ID[sn] := iid;

        writedat;

    end;

end;

end.

```

Medium and High altitude satellite support requests sample data sets generation code.

This code was developed to generate medium and high altitude satellite support request sample data sets. The code is shown below:

```

Program hreqrn;
Uses winCrt;
Var

    STz :array[1..14] of integer;
    STn :array[1..14] of integer;
    RT : array[1..33] of integer;
    SV : array[1..5000] of longint;
    ID : array[1..5000] of longint;
    nstz:array[1..14] of integer;
    s,J,k:integer;
    TM,TR:integer;
    l:integer;
    r,EV:integer;
    tmp,gap:integer;
    used,true,false,isv,iid:integer;

    procedure writedat;

```

Begin

If s=1 then

Begin

```
writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
```

I:=I+1;

end;

If s=2 then

Begin

```
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;
```

If s=3 then

Begin

```
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;
```

If s=4 then

Begin

```
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
```

```

end;

If s=5 then
Begin

writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

If s=6 then
Begin

writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

If s=7 then
Begin

writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

If s=8 then
Begin

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);

```

```

writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=9 then
Begin

```

```

writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=10 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=11 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);

```

```

writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

end;
procedure writedat1;
begin

  If s=12 then
  Begin

    writeln(output,I:4,'','TNDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
    writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
    writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
    writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
    I:=I+1;
    end;

    If s=13 then
    Begin

      writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
      writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
      writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
      writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
      writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
      writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
      writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
      I:=I+1;
      end;

      If s=14 then
      Begin

        writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
        writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
        writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
        writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
        writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
        writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
        I:=I+1;
        end;

        If s=15 then
        Begin

          writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
          writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
          writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
          writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
          writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
          writeln(output,I:4,'','TNDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
          writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
          writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);

```

```

writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=16 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=17 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=18 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);

```

```

writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=19 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=20 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=21 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);

```

```
I:=I+1;  
end;
```

```
If s=22 then  
Begin
```

```
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
I:=I+1;  
end;
```

```
If s=23 then  
Begin
```

```
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
I:=I+1;  
end;  
end;
```

```
procedure writedat2;  
begin
```

```
If s=24 then  
Begin
```

```
writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
I:=I+1;  
end;
```

```
If s=25 then  
Begin
```

```
writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);  
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
```

```

writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=26 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=27 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=28 then
Begin

```

```

writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);

```

```
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;
```

```
If s=29 then
Begin
```

```
writeln(output,I:4,'','POGO-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','POGO-C',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','HULA-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;
```

```
If s=30 then
Begin
```

```
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;
```

```
If s=31 then
Begin
```

```
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','INDI-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;
```

```
If s=32 then
Begin
```

```
writeln(output,I:4,'','COOK-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','COOK-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','BOSS-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','LION-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
```

```

writeln(output,I:4,'','PIKE-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','REEF-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;

```

```

If s=33 then

```

```

Begin

```

```

writeln(output,I:4,'','GUAM-A',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
writeln(output,I:4,'','GUAM-B',SV[I]:5,EV:5,r:5,' 15 ',ID[I]);
I:=I+1;
end;
end;

```

```

Begin

```

```

Assign(output,'A:\hr.dat');

```

```

Rewrite(Output);

```

```

STz[1]:=5; STz[2]:=10; STz[3]:=15; STz[4]:=20; STz[5]:=25; STz[6]:=30;
STz[7]:=35; STz[8]:=40; STz[9]:=45; STz[10]:=50; STz[11]:=165; STz[12]:=300;
STz[13]:=435; STz[14]:=660;

```

```

true:=1;

```

```

false:=0;

```

```

gap:= 0;

```

```

repeat

```

```

Randomize;

```

```

STn[1] := round(4*random+22); {from 22 to 26}
STn[2] := round(5*random+39); {from 39 to 44}
STn[3] := round(7*random+30); {from 30 to 37}
STn[4] := round(10*random+25); {from 25 to 35}
STn[5] := round(3*random+5); {from 5 to 8}

```

```

STn[6] := round(2*random+2); {from 2 to 4}
STn[7] := round(2*random+5); {from 5 to 7}
STn[8] := round(2*random+3); {from 3 to 5}
STn[9] := round(6*random+6); {from 6 to 12}
STn[10] := round(2*random+2); {from 2 to 4}
STn[11] := round(2*random); {from 0 to 2}
STn[12] := round(2*random); {from 0 to 2}
STn[13] := round(2*random); {from 0 to 2}
STn[14] := round(random); {from 0 to 1}

```

```

TM:=0;

```

```

for s:=1 to 14 do

```

```

begin

```

```

TM:=TM+STn[s];

```

```

end;

```

```

RT[1]:=round(TM*0.101);

```

```

RT[2]:=round(TM*0.089);

```

```

RT[3]:=round(TM*0.083);
RT[4]:=round(TM*0.077);
RT[5]:=round(TM*0.071);
RT[6]:=round(TM*0.071);
RT[7]:=round(TM*0.065);
RT[8]:=round(TM*0.042);
RT[9]:=round(TM*0.042);
RT[10]:=round(TM*0.036);
RT[11]:=round(TM*0.036);
RT[12]:=round(TM*0.030);
RT[13]:=round(TM*0.030);
RT[14]:=round(TM*0.030);
RT[15]:=round(TM*0.024);
RT[16]:=round(TM*0.024);
RT[17]:=round(TM*0.018);
RT[18]:=round(TM*0.012);
RT[19]:=round(TM*0.012);
RT[20]:=round(TM*0.012);
RT[21]:=round(TM*0.012);
RT[22]:=round(TM*0.012);
RT[23]:=round(TM*0.012);
RT[24]:=round(TM*0.006);
RT[25]:=round(TM*0.006);
RT[26]:=round(TM*0.006);
RT[27]:=round(TM*0.006);
RT[28]:=round(TM*0.006);
RT[29]:=round(TM*0.006);
RT[30]:=round(TM*0.006);
RT[31]:=round(TM*0.006);
RT[32]:=round(TM*0.006);
RT[33]:=round(TM*0.006);

TR:=0;
for s:=1 to 33 do
begin
  TR:=TR+RT[s];
end;

until TM=TR;

for s:=1 to 14 do
begin
  nstz[s]:=0;
end;

l:=1;

for s:=1 to 33 do
begin
  for J:=1 to RT[s] do
    begin

```

```

used:=false;
while (used=false) do
  begin
    tmp:=round(13*random+1);

    if (nstz[tmp]<STn[tmp]) then
      used:=true;
    end;

R:=stz[tmp];
nstz[tmp]:=nstz[tmp]+1;

used := true;
  while (used = true) do
    begin
      isv:=round((1440*random)/5)*5;
      if (R=5) then gap:=60;
      if (R=10) then gap:=40;
      if (R=15) then gap:=35;
      if (R=20) then gap:=60;
      if (R=25) then gap:=30;
      if (R=30) then gap:=10;
      if (R=35) then gap:=20;
      if (R=40) then gap:=10;
      if (R=45) then gap:=5;
      if (R=50) then gap:=0;
      if (R=165) then gap:=0;
      if (R=300) then gap:=0;
      if (R=435) then gap:=0;
      if (R=660) then gap:=0;

      EV := isv + gap + R;

      used := false;
      for k:= 1 to I do
        IF (isv = SV[k]) or (EV>1440) then
          used := true;

      end;
      SV[I] := isv;

used := true;
    while (used = true) do
      begin
        iid:=round(8000*random+1000);

        used := false;
        for k:= 1 to I do
          IF (iid = ID[k]) then
            used := true;

        end;
        ID[I] := iid;

```

```
if (s<12) then writedat;  
if (s>=12) and (s<24) then writedat1;  
if (s>=24) then writedat2;
```

```
end;
```

```
end;
```

```
end.
```

APPENDIX B

Sort Program.

This FORTRAN program sorts the data sets by ascending order of starting visibility time.

Sort for Low altitude satellite.

```
PROGRAM SORTlow
```

```
INTEGER SNUM(1000)
CHARACTER*6 GTS(1000)
INTEGER BV(1000), EV(1000), REQ(1000), TA(1000), IDENT(1000)
REAL REV(1000)
```

```
INTEGER MAXLINE, NUMLINE, MAXNUMBER
INTEGER SN, SNIDX
INTEGER NEWSNUM
```

```
OPEN(UNIT=10,FILE='lr27.dat',STATUS='old')
OPEN(UNIT=20,FILE='nlr27.dat',STATUS='new')
```

```
MAXLINE = 1000
NUMLINE = 0
MAXNUMBER = 99999
```

```
C
C READ IN DATA FROM ASCII FILE
C
```

```
DO I = 1, MAXLINE
```

```
    READ(10,FMT=90, END=400) SNUM(I),GTS(I),bv(I),ev(I),req(I),
    * TA(I),IDENT(I),REV(I)
90  FORMAT(I4,1X,A6,1X,I4,I5,I5,1X,I2,1X,I4,1X,F6.1)
    NUMLINE = NUMLINE + 1
END DO
```

```
C
C ASCENDING SORT BY BV
C
400 DO I = 1, NUMLINE-1
```

SN = MAXNUMBER

SNUM(NUMLINE+1) = SNUM(I)
GTS(NUMLINE+1) = GTS(I)
BV(NUMLINE+1) = BV(I)
EV(NUMLINE+1) = EV(I)
REQ(NUMLINE+1) = REQ(I)
TA(NUMLINE+1) = TA(I)
IDENT(NUMLINE+1) = IDENT(I)
REV(NUMLINE+1) = REV(I)

DO J = 1, NUMLINE
IF (SN.GT.BV(J)) THEN
SN = BV(J)
SNIDX = J
ENDIF
END DO

SNUM(I) = SNUM(SNIDX)
GTS(I) = GTS(SNIDX)
BV(I) = BV(SNIDX)
EV(I) = EV(SNIDX)
REQ(I) = REQ(SNIDX)
TA(I) = TA(SNIDX)
IDENT(I) = IDENT(SNIDX)
REV(I) = REV(SNIDX)

SNUM(SNIDX) = SNUM(NUMLINE+1)
GTS(SNIDX) = GTS(NUMLINE+1)
BV(SNIDX) = BV(NUMLINE+1)
EV(SNIDX) = EV(NUMLINE+1)
REQ(SNIDX) = REQ(NUMLINE+1)
TA(SNIDX) = TA(NUMLINE+1)
IDENT(SNIDX) = IDENT(NUMLINE+1)
REV(SNIDX) = REV(NUMLINE+1)

END DO

C
C WRITE DATA TO ASCII FILE
C

NEWSNUM = 0

DO I = 1, NUMLINE

IF (I.NE.1 .AND. SNUM(I-1).NE.SNUM(I)) THEN
NEWSNUM = NEWSNUM + 1
ENDIF

```

      IF (NEWSNUM .NE. 0) THEN
        WRITE(20,FMT=91) NEWSNUM,GTS(I),bv(I),ev(I),req(I),
      *  TA(I),IDENT(I),REV(I)
90    FORMAT(I4,1X,A6,1X,I4,I5,I5,1X,I2,1X,I4,'0',F6.1)
      ENDIF
    END DO

```

```

STOP
END

```

Sort for Medium and high altitude satellite.

```

PROGRAM SORThigh
INTEGER SNUM(5000)
CHARACTER*6 GTS(5000)
INTEGER BV(5000), EV(5000), REQ(5000), TA(5000), IDENT(5000)

INTEGER MAXLINE, NUMLINE, MAXNUMBER
INTEGER SN, SNIDX
INTEGER NEWSNUM

OPEN(UNIT=10,FILE='hhh1.dat',STATUS='old')
OPEN(UNIT=20,FILE='nhhh1.dat',STATUS='new')

MAXLINE = 5000
NUMLINE = 0
MAXNUMBER = 99999

C
C READ IN DATA FROM ASCII FILE
C
  DO I = 1, MAXLINE

    READ(10,FMT=90, END=400) SNUM(I),GTS(I),bv(I),ev(I),req(I),
  *  TA(I),IDENT(I)
90    FORMAT(I4,1X,A6,1X,I4,I5,I5,1X,I2,1X,I4)
    NUMLINE = NUMLINE + 1
  END DO

C
C ASCENDING SORT BY BV
C
400  DO I = 1, NUMLINE-1

    SN = MAXNUMBER

    SNUM(NUMLINE+1) = SNUM(I)
    GTS(NUMLINE+1) = GTS(I)

```

```

    BV(NUMLINE+1) = BV(I)
    EV(NUMLINE+1) = EV(I)
    REQ(NUMLINE+1) = REQ(I)
    TA(NUMLINE+1) = TA(I)
    IDENT(NUMLINE+1) = IDENT(I)

DO J = 1, NUMLINE
    IF ( SN.GT.BV(J) ) THEN
        SN = BV(J)
        SNIDX = J
    ENDIF
END DO

    SNUM(I) = SNUM(SNIDX)
    GTS(I) = GTS(SNIDX)
    BV(I) = BV(SNIDX)
    EV(I) = EV(SNIDX)
    REQ(I) = REQ(SNIDX)
    TA(I) = TA(SNIDX)
    IDENT(I) = IDENT(SNIDX)

    SNUM(SNIDX) = SNUM(NUMLINE+1)
    GTS(SNIDX) = GTS(NUMLINE+1)
    BV(SNIDX) = BV(NUMLINE+1)
    EV(SNIDX) = EV(NUMLINE+1)
    REQ(SNIDX) = REQ(NUMLINE+1)
    TA(SNIDX) = TA(NUMLINE+1)
    IDENT(SNIDX) = IDENT(NUMLINE+1)

END DO

C
C WRITE DATA TO ASCII FILE
C
    NEWSNUM = 0

    DO I = 1, NUMLINE

        IF (I.NE.1 .AND. SNUM(I-1).NE.SNUM(I) ) THEN
            NEWSNUM = NEWSNUM + 1
        ENDIF

        IF (NEWSNUM .NE. 0) THEN
            WRITE(20,FMT=91) NEWSNUM,GTS(I),bv(I),ev(I),req(I),
            * TA(I),IDENT(I)
91    FORMAT(I4,1X,A6,1X,I4,I5,I5,1X,I2,1X,I4,'00000.0')
        ENDIF
    END DO

```

STOP
END

APPENDIX C

Schalck's SRS algorithm programs.

This appendix details Schalck's SRS algorithm which was used to generate a single day's 24 hour schedule. Most of the programs are revisions of Schalck's work.

IPLINK.FOR This FORTRAN program is executed on the PCDX486. It completes the pre-processing required for input into the GAMS program.

```
PROGRAM iplinkhigh
  INTEGER I,J,K,REQ,IREV,SNUM
  CHARACTER*3 GGTS(18)
  CHARACTER*6 GTS,AGS(18)
  INTEGER TA,BV,EV,IDENT,csum
  INTEGER TO(400),NWI(400)
  REAL REV,W(400)
  INTEGER R(400,18), X(400,18),ABV(400,18),AEV(400,18)
  INTEGER CNT, CNT1
  IREV=0.0
  OPEN(UNIT=10,FILE='sup.dat',STATUS='UNKNOWN')
  OPEN(UNIT=21,FILE='req.dat',STATUS='UNKNOWN')
  OPEN(UNIT=12,FILE='ntable.dat',STATUS='UNKNOWN')
  OPEN(UNIT=22,FILE='avg.dat',STATUS='UNKNOWN')
  OPEN(UNIT=9,FILE='hold.dat',STATUS='UNKNOWN')

  I=0
  IDENT=0
  AGS(1)='POGO-A'
  AGS(2)='POGO-B'
  AGS(3)='POGO-C'
  AGS(4)='POGO-D'
  AGS(5)='HULA-A'
  AGS(6)='HULA-B'
  AGS(7)='COOK-A'
  AGS(8)='COOK-B'
  AGS(9)='INDI-A'
  AGS(10)='INDI-B'
  AGS(11)='BOSS-A'
  AGS(12)='BOSS-B'
```

```

    AGS(13)='LION-A'
    AGS(14)='LION-B'
    AGS(15)='GUAM-A'
    AGS(16)='GUAM-B'
    AGS(17)='PIKE-A'
    AGS(18)='REEF-A'
    GGTS(1)='P-A'
    GGTS(2)='P-B'
    GGTS(3)='P-C'
    GGTS(4)='P-D'
    GGTS(5)='H-A'
    GGTS(6)='H-B'
    GGTS(7)='C-A'
    GGTS(8)='C-B'
    GGTS(9)='I-A'
    GGTS(10)='I-B'
    GGTS(11)='B-A'
    GGTS(12)='B-B'
    GGTS(13)='L-A'
    GGTS(14)='L-B'
    GGTS(15)='G-A'
    GGTS(16)='G-B'
    GGTS(17)=' PI'
    GGTS(18)='REF'
    DO 11 J=1,400
    DO 12 K=1,18
        NWI(J)=0
        X(J,K)=0
        ABV(J,K)=0
        AEV(J,K)=0
        W(J)=0
        R(J,K)=0
        TO(J)=0
12  CONTINUE
11  CONTINUE
C
C  READ IN DATA FROM ASCII FILE
C
10  READ(21,FMT=98,ERR=200,END=200) SNUM,GTS,bv,ev,req,
    * TA,IDENT,REV
98  FORMAT(I4,1X,A6,1X,I5,I5,I5,1X,I2,1X,I4,1X,F6.1)

    PRINT *, 'SNUM = ',SNUM,' I = ',I

    K=0
    DO 35 J = 1,18
        IF (GTS .EQ. AGS(J)) THEN
            K=J
        ENDIF
35  CONTINUE
    IF (K .EQ. 0) GOTO 10

```

```

C   IF (REQ .lt. 10) GOTO 10

    FLAG=0
    cnt=1
    cnt1=1
    DO 23 J = 1,400
      IF (NWI(J) .GT. 0) CNT1=CNT1+1

      IF (SNUM .EQ. NWI(J)) THEN
        CNT=J
        FLAG=1
      ENDIF
23  CONTINUE

    IF ( FLAG .EQ. 1) THEN
      I=CNT
    ELSE
      I=CNT1
      NWI(I)=SNUM
    ENDIF

    WRITE(10,99) SNUM,I,GTS
99  FORMAT(I6,2X,I6,1X,A6)
    K=0
    DO 25 J = 1,18
      IF (GTS .EQ. AGS(J)) THEN
        K=J
      ENDIF
25  CONTINUE
    IF (K .EQ. 0) GOTO 10
    X(I,K)=1
    ABV(I,K)=BV
    AEV(I,K)=EV
    TO(I)=15
    R(I,K)=REQ

    IF (I .GE. 85) THEN
      WRITE(9,FMT=98) SNUM,GTS,bv,ev,req,
* TA,IDENT,REV
      GOTO 10

    ENDIF

C
C READ NEXT RECORD
C
    GOTO 10
C
C CLOSE FILES
C
C
C DETERMINE WEIGHT OF VARIABLE
200 Csum=0

```

```

DO 22 J= 1,I
SUM=0
DO 21 K=1,18
SUM=SUM+X(J,K)
21 CONTINUE
IF (SUM .GT. 0) THEN
  W(J)=1/sum
ENDIF
csum=csum+sum
22 CONTINUE
write(22,97) i,csum
97 FORMAT(I5,2X, I8)

C
C WRITE TABLES TO FILE
C CREATE GTS HEADING
C

WRITE(12,102)
102 FORMAT(5X,'SETS')
103 FORMAT(' I supports /1*',I4, '/')
WRITE(12,103) I
WRITE (12,900)
900 FORMAT(' ')
WRITE (12,105)
105 FORMAT(7X,'J GTS')
DO 19 J = 1,18
IF (J .EQ. 1) THEN
  WRITE(12,106) GGTS(1)
106 FORMAT(8X, '/', A3)
ENDIF
IF (J .EQ. 18) THEN
  WRITE(12,110) GGTS(18)
110 FORMAT(10X, A3, ',')
ENDIF
IF (J .GT. 1 .AND. J .LT. 18) THEN
  WRITE(12,107) GGTS(J)
107 FORMAT(10X, A3)
ENDIF
19 CONTINUE
WRITE (12,900)
WRITE(12,104)
104 FORMAT(2X,'ALIAS(I,H);')
WRITE(12,900)
WRITE(12,991)
991 FORMAT('SET OFFDIAG(I,H);')
WRITE(12,992)
992 FORMAT('OFFDIAG(I,H)=YES$(ORD(I) ne ORD(H))$(ORD(I) LT ORD(H));')
WRITE(12,900)
WRITE(12,997)
997 FORMAT('SCALAR M large positive constant /5000/;')
WRITE (12,900)
WRITE (12,201)

```

```

201  FORMAT('PARAMETERS')
      WRITE (12,900)
      WRITE(12,202)
202  FORMAT(' W(I) weight of a support')
      DO 49 J = 1 ,I
        IF (J .EQ. 1) THEN
          WRITE (12,203) J, W(J)
203    FORMAT(4X,' ',I4,2X,F4.2)
        ENDIF
        IF (J .GT. 1 .AND. J .LT. I) THEN
          WRITE (12,204) J, W(J)
204    FORMAT(5X,I4,2X,F4.2)
        ENDIF
        IF (J .EQ. I) THEN
          WRITE (12,205) J, W(J)
205    FORMAT(5X,I4,2X,F4.2,' /')
        ENDIF
49  CONTINUE
      WRITE (12,900)
      WRITE(12,302)
302  FORMAT(' TO(I) turnaround time of a support')
      DO 59 J = 1 ,I
        IF (J .EQ. 1) THEN
          WRITE (12,303) J, TO(J)
303    FORMAT(4X,' ',I4,2X,I3)
        ENDIF
        IF (J .GT. 1 .AND. J .LT. I) THEN
          WRITE (12,304) J,TO(J)
304    FORMAT(5X,I4,2X,I3)
        ENDIF
        IF (J .EQ. I) THEN
          WRITE (12,305) J,TO(J)
305    FORMAT(5X,I4,2X,I3,' /;')
        ENDIF
59  CONTINUE
      WRITE(12,900)
      WRITE(12,900)
      WRITE(12,402)
402  FORMAT('TABLE R(I,J) request length supports of i and j')
      WRITE(12,900)
      DO 48 K=1,1
        WRITE(12,411) (GGTS(J),J=1,11)
411    FORMAT(5X,11A6)
48  CONTINUE
      DO 45 J = 1, I
        WRITE(12,409) J,(R(J,K),K=1,11)
45  CONTINUE
409  FORMAT(I4,1X,11I6)
      DO 46 K=1,1
        WRITE(12,421) (GGTS(J),J=12,18)
421    FORMAT(3X,'+',1X,7A6)
46  CONTINUE
      DO 47 J = 1, I

```

```

        WRITE(12,419) J,(R(J,K),K=12,18)
47  CONTINUE
419  FORMAT(I4,1X,7I6)
        WRITE (12,401)
401  FORMAT(2X,';')
        WRITE(12,900)
        WRITE(12,108)
108  FORMAT('TABLE N(I,J) feasible supports of i and j')
        WRITE(12,900)
        DO 28 K=1,1
            WRITE(12,111) (GGTS(J),J=1,11)
111  FORMAT(5X,11A6)
28  CONTINUE
        DO 29 J = 1, I
            WRITE(12,109) J,(X(J,K),K=1,11)
29  CONTINUE
109  FORMAT(I4,1X,11I6)
        DO 38 K=1,1
            WRITE(12,121) (GGTS(J),J=12,18)
121  FORMAT(3X,+',1X,7A6)
38  CONTINUE
        DO 39 J = 1, I
            WRITE(12,119) J,(X(J,K),K=12,18)
39  CONTINUE
119  FORMAT(I4,1X,7I6)
        WRITE (12,901)
901  FORMAT(2X,';')
        WRITE(12,900)
        WRITE(12,508)
508  FORMAT('TABLE BV(I,J) beginning of visibility')
        WRITE(12,900)
        DO 52 K=1,1
            WRITE(12,511) (GGTS(J),J=1,11)
511  FORMAT(5X,11A6)
52  CONTINUE
        DO 53 J = 1, I
            WRITE(12,509) J,(ABV(J,K),K=1,11)
53  CONTINUE
509  FORMAT(I4,1X,11I6)
        WRITE(12,900)
        DO 62 K=1,1
            WRITE(12,521) (GGTS(J),J=12,18)
521  FORMAT(3X,+',1X,7A6)
62  CONTINUE
        DO 63 J = 1, I
            WRITE(12,519) J,(ABV(J,K),K=12,18)
63  CONTINUE
519  FORMAT(I4,1X,7I6)
        WRITE (12,901)
        WRITE (12,900)
        WRITE(12,608)
608  FORMAT('TABLE EV(I,J) ending of visibility')
        WRITE(12,900)

```

```

DO 78 K=1,1
  WRITE(12,611) (GGTS(J),J=1,11)
611  FORMAT(5X,11A6)
78  CONTINUE
  DO 79 J = 1, I
    WRITE(12,709) J,(AEV(J,K),K=1,11)
79  CONTINUE
709  FORMAT(I4,1X,11I6)
    WRITE(12,900)
    DO 88 K=1,1
      WRITE(12,721) (GGTS(J),J=12,18)
721  FORMAT(3X,'+',1X,7A6)
88  CONTINUE
    DO 89 J = 1, I
      WRITE(12,719) J,(AEV(J,K),K=12,18)
89  CONTINUE
719  FORMAT(I4,1X,7I6)
    WRITE (12,901)
    write(12,900)
    WRITE(12,994)
994  FORMAT('SET D(I,H,J);')
    WRITE(12,995)
995  FORMAT('D(I,H,J)=YES$N(H,J)$N(I,J)$OFFDIAG(I,H)$ (EV(I,J) GT
    *(BV(H,J)-TO(H));')
412  FORMAT('SET D4(I,H,J);')
415  FORMAT('SET D5(I,H,J);')
410  FORMAT('SET D6(I,H,J);')
416  FORMAT('D4(I,H,J)=YES$D1(I,H,J)$ (NOT(D2(I,H,J)));')
408  FORMAT('D5(I,H,J)=YES$ (NOT(D1(I,H,J)))$D2(I,H,J);')
407  FORMAT('D6(I,H,J)=YES$D1(I,H,J)$D2(I,H,J);')
999  FORMAT('SET D1(I,H,J);')
993  FORMAT('SET D2(I,H,J);')
413  FORMAT('SET D3(I,H,J);')
996  FORMAT('D1(I,H,J)=YES$D(I,H,J)$((BV(H,J)+R(H,J)+TO(I)+R(I,J))
    *LT(EV(I,J)));')
414  FORMAT('D2(I,H,J)=YES$D(I,H,J)$((BV(I,J)+R(I,J)+TO(H)+R(H,J))
    *LT(EV(H,J)));')
998  FORMAT('D3(I,H,J)=YES$D(I,H,J)$ (NOT(D1(I,H,J)))$ (NOT(D2(I,H,J)))
    *,')
    WRITE(12,999)
    WRITE(12,996)
    WRITE(12,993)
    WRITE(12,414)
    WRITE(12,413)
    WRITE(12,998)
    WRITE(12,412)
    WRITE(12,416)
    WRITE(12,415)
    WRITE(12,408)
    WRITE(12,410)
    WRITE(12,407)
    WRITE(12,900)
    CLOSE(12)

```

```
CLOSE(10)
CLOSE(21)
```

```
300 PRINT *, 'NUMBER OF SUPPORT IS', I
END
```

SRS.GMS. This is the GAMS program which models the mixed integer program (MIP) formulation and calls the MIP solver

```
$include "ntable.dat"
sets jj(j) dynamic subset of i to hold columns for subtable
      cc(j) dynamic subset of i to hold unprinted columns
      s subtables / 1*12 /;
scalar maxcol;
```

VARIABLES

```
ST(I,J) start time for support i at GTS j
X(I,J) support i and GTS j 1 if support occurs 0 otherwise
Y(H,I,J) relax or enforce constraint for supports h i and GTS j
Z total weighted number of supports scheduled;
```

```
POSITIVE VARIABLE ST;
BINARY VARIABLES X,Y;
```

EQUATIONS

```
SCH obj function - weighted number of supports scheduled
SUPONE(I) schedule support only once
BEGSUP(I,J) schedule support after its beginning visibility
ENDSUP(I,J) schedule support before its end of visibility
NCCSUP1(J,I,H) no concurrent supports on a GTS j ST_i lt ST_j
RELAX1(J,I,H) relax or enforce no concurrent support constraints
RELAX2(J,I,H) relax or enforce no concurrent support constraints
NCCSUP2(J,I,H) no concurrent supports ST i GT ST h
SUP12(J,I,H)
SUP21(J,I,H)
NSUP(J,I,H);
```

```
SCH.. Z=E= SUM((I,J)$N(I,J), X(I,J));
```

```
SUPONE(I).. SUM(J$N(I,J), X(I,J)) =L= 1;
```

```
BEGSUP(I,J)$N(I,J).. ST(I,J)=G= BV(I,J);
```

```
ENDSUP(I,J)$N(I,J).. ST(I,J)=L= EV(I,J)-R(I,J);
```

```
SUP21(J,I,H)$D4(I,H,J).. ST(I,J)=G= ST(H,J)+R(H,J)
```

```

+TO(I)-M*(1-X(I,J))-M*(1-X(H,J));

SUP12(J,I,H)$D5(I,H,J).. ST(I,J)+R(I,J)+TO(H)
=L= ST(H,J) +M*(1-X(I,J))+M*(1-X(H,J));

NCCSUP1(J,I,H)$D6(I,H,J).. ST(I,J) + R(I,J)
+ TO(H) =L= ST(H,J) +M*Y(H,I,J) + M*(1-X(I,J))
+ M*(1-X(H,J));

RELAX1(J,I,H)$D6(I,H,J).. ST(I,J) - ST(H,J) =L= M*Y(H,I,J);

RELAX2(J,I,H)$D6(I,H,J).. ST(H,J) - ST(I,J)
+ .5=L= M*(1-Y(H,I,J));

NCCSUP2(J,I,H)$D6(I,H,J).. ST(I,J) =G= ST(H,J) + R(H,J)
+ TO(I)-M*(1-Y(H,I,J))-M*(1-X(I,J))-M*(1-X(H,J));

NSUP(J,I,H)$D3(I,H,J).. X(I,J)+X(H,J) =L= 1;
MODEL SCHEDULE SRS Scheduling Solution /ALL/;

OPTION ITERLIM=5000000;

OPTION RESLIM=100000;

OPTION OPTCR=0.05;

SOLVE SCHEDULE USING MIP MAXIMIZING Z;

DISPLAY ST.L, X.L;

file res /sch.dat/;

res.pw = 78;
put res ' table x(i,j) this is a table of scheduled supports';
jj(j) = no;
cc(j) = yes;
loop(s$card(cc),
maxcol=floor(res.pw/7-1);
loop(cc$maxcol,
jj(cc) = yes;
maxcol=maxcol-1);
if((card(cc) ne card(jj)), put // '+':6);
if((not(card(cc) ne card(jj))), put // ' ':6 );
loop(jj, put jj.tl:>5); put /;
loop(i,
put / i.tl:5;
loop(jj, put x.L(i,jj):5:1) );
cc(jj) = no;
jj(jj) = no );
put$card(cc)// '**** more than ' card(s):0:0 ' subtables'
/ '**** ' card(cc):0:0 ' columns not written';

```

```

abort$card(cc) 'not all columns were printed' , cc;

file res1 /strt.dat/;

res1.pw = 78;

put res1 ' table st(i,j) this is a table of support start times/';
jj(j) = no;
cc(j) = yes;
loop(s$card(cc),
  maxcol=floor(res1.pw/7-1);
  loop(cc$maxcol,
    jj(cc) = yes;
    maxcol=maxcol-1);
  if((card(cc) ne card(j)), put // '+':6);
  if((not(card(cc) ne card(j))), put // ' ':6 );
  loop(jj, put jj.tl:>5); put /;
  loop(i,
    put / i.tl:5;
    loop(jj, put st.L(i,jj):5:0) );
  cc(jj) = no;
  jj(jj) = no );
put$card(cc) // '**** more than ' card(s):0:0 ' subtables'
/ '**** ' card(cc):0:0 ' columns not written';
abort$card(cc) 'not all columns were printed' , cc;

```

```

file res2 /req.dat/;

res2.pw = 78;

put res2 ' table r(i,j) this is a table of support request times/';
jj(j) = no;
cc(j) = yes;
loop(s$card(cc),
  maxcol=floor(res2.pw/7-1);
  loop(cc$maxcol,
    jj(cc) = yes;
    maxcol=maxcol-1);
  if((card(cc) ne card(j)), put // '+':6);
  if((not(card(cc) ne card(j))), put // ' ':6 );
  loop(jj, put jj.tl:>5); put /;
  loop(i,
    put / i.tl:5;
    loop(jj, put R(i,jj):5:0) );
  cc(jj) = no;
  jj(jj) = no );
put$card(cc) // '**** more than ' card(s):0:0 ' subtables'
/ '**** ' card(cc):0:0 ' columns not written';
abort$card(cc) 'not all columns were printed' , cc;

```

SCHUP.PAS. Schedule update. This program takes the output from *SRS.GMS* (SCH.DAT and STRT.DAT) and an output from *IPLINK.FOR* (SUP.DAT) and adds it to previously scheduled activities to generate an updated schedule.

```

program test;
uses WinCrt;
  VAR sch: ARRAY [1..18] of string[5];
      strt: ARRAY [1..18] of string[5];

      strn: ARRAY [1..18] of integer;

      SN: ARRAY [1..85] OF INTEGER;
      I,j,CNT,supn,holdn,ACTN,lo1n,lo2n,act1n,stnn,strttn,strttn,endn,stn, NUM: integer;
      hfn,bvn,evn,durn,tatn,hfn1,bvn1,evn1,durn1,tatn1 : integer;
      waste: string[80];
      GTS,gts1: STRING[7];
      fill: string[11];
      sup: string[6];
      fill1: string[11];
      hold: string[8];
      lo2,lo1,b : string[2];
      act,bs : string[3];
      act1: string[4];
      INFILE,infile2,outfile,outfile1,outfile3,srsfile,outfile2,infile1 : TEXT;
TYPE
  GRND= STRING[6];

  SRS= RECORD

    GTS : GRND;
    BV : INTEGER;
    EV : INTEGER;
    REQ : INTEGER;
    TOT : INTEGER;
    IRON: INTEGER;
    REV : REAL;
  END;

VAR
  ABV,AEV,TA,RE,ID,SU : INTEGER;
  REVV : REAL;
  TGTS: GRND;
  CH: STRING[1];

BEGIN
  WRITELN('WHERE IS OPT FILE FROM IE NDAYLF');
  CNT:=0;
  ASSIGN(SRSFILE,'a:\sch.DAT');
  RESET(SRSFILE);
  ASSIGN(INFILE,'a:\strt.dat');
  RESET(INFILE);

```

```

assign(infile1,'a:\Reqlf.dat');
reset(infile1);
assign(infile2,'a:\sup.dat');
reset(infile2);
assign(outfile,'a:\trash1.dat');
assign(outfile1,'a:\fs.dat');
assign(outfile2,'a:\trash2.dat');
assign(outfile3,'a:\fs1.dat');

```

```

WHILE NOT EOF(SRSFILE) DO
BEGIN
  for i:= 1 to 16 do
    READ(SRSFILE,sch[i]);
  for i:= 1 to 16 do
    READ(infile,strt[i]);
    readln(srsfile,sch[17]);
    readln(infile,strt[17]);
    for i:= 1 to 16 do
      val(strt[i],strn[i],j);
    IF sch[1]='1 ' THEN cnt:=cnt+1;

  IF CNT=1 THEN
  BEGIN
    rewrite(outfile);
    FOR i:=2 TO 11 DO
      if sch[i]=' 1.0'then writeln(outfile,sch[1],i-1,strn[i]:5);
    end;
    if cnt=2 then
    begin
      for i:=2 TO 11 DO
        if sch[i]=' 1.0'then writeln(outfile,sch[1],(i+9),strn[i]:5);
      end;

    end;
    reset(outfile);
    for cnt:=1 to 200 do
    begin
      reset(infile2);
      repeat
        readln(infile2,sup,hold,fill);
        val(hold,holdn,j);
        val(sup,supn,j);
      until (holdn>=cnt)or(eof(infile2));
      rewrite(outfile2);
      repeat
        readln(outfile2,strn[2],fill1);

        if cnt=strn[2] then writeln(outfile2,supn:5,fill1);
        until eof(outfile2);
      end;

      rewrite(outfile1);

```

```

    cnt:=0;
    reset(outfile2);
    repeat
    readln(outfile2,actn,stn,strtn);

        if(cnt=0) or (cnt<actn) then cnt:=actn;

    reset(infile1);
    repeat
    readln(infile1,act1,gts,strn[1],strn[2],strn[3],fill);

    val(act1,act1n,i);
    until (actn=act1n) or eof(infile1);

    if stn=1 then gts:=' POGO-A ';
    if stn=2 then gts:=' POGO-B ';
        if stn=3 then gts:=' POGO-C ';
    if stn=5 then gts:=' HULA-A ';
        if stn=6 then gts:=' HULA-B ';
    if stn=7 then gts:=' COOK-A ';
        if stn=8 then gts:=' COOK-B ';
    if stn=9 then gts:=' INDI-A ';
        if stn=10 then gts:=' INDI-B ';
    if stn=11 then gts:=' BOSS-A ';
        if stn=12 then gts:=' BOSS-B ';
    if stn=13 then gts:=' LION-A ';
        if stn=14 then gts:=' LION-B ';
    if stn=15 then gts:=' GUAM-A ';
        if stn=16 then gts:=' GUAM-B ';
    if stn=17 then gts:=' PIKE-A ';
    if stn=18 then gts:=' REEF-A ';
    if cnt=actn then
        BEGIN

        endn:=strtn+strn[3];
        WRITELN(OUTFILE1,actn:3,gts,strtn:5,endn:5,strn[3]:5,FILL);

        CNT:=actn+1;
        END;

    until eof(outfile2);
    reset(outfile1);
    rewrite(outfile2);
repeat
readln(outfile1,hfn,gts,bvn,evn,durn,tatn,fill);
    writeln(outfile2,hfn:4,gts,bvn:5,evn:5,durn:5,tatn:3,fill);
    until eof(outfile1);
    repeat
    readln(outfile1,hfn1,gts1,bvn1,evn1,durn1,tatn1,fill1);
    writeln(outfile2,hfn1:4,gts1,bvn1:5,evn1:5,durn1:5,tatn1:3,fill1);
    until eof(outfile1);
    rewrite(outfile3);

```

```

reset(outfile2);
  for i:=-60 to 1500 do
  begin
    reset(outfile2);
    repeat
      readln(outfile2,hfn,gts,bvn,evn,durn,tatn,fill);

      if bvn=i then writeln(outfile3,hfn:4,gts,bvn:5,evn:5,durn:5,tatn:3,fill);

    until eof(outfile2);
  end;
  reset(outfile);
  repeat
    readln(outfile);
  until eof(outfile);
  reset(outfile2);
  repeat
    readln(outfile2);
  until eof(outfile2);
  reset(outfile3);
  repeat
    readln(outfile3);
  until eof(outfile3);
end.

```

SUBSCH.PAS. Subtract schedule. This PASCAL program takes scheduled activities and subtracts the scheduled time block from that particular RTS side's support request possibilities in the block of requests about to be scheduled.

```

program subsch;
uses WinCrt;
Type
  mat = array[1..40, 1..3] of Integer;

Var
  hfn,flg,bvn,evn,durn,tatn,hfn1,bvn1,bsn1,evn1,bvnh,evnh,durn1,tatn1: integer;

  gts,gts1: string[7];

  fill : string[10];
  fill1 : string[10];
  revv,revlf,revhf :real;
  dum:STRING[9];
  Infile,Infile1,outfile : Text;
  stats : mat;
  Match :boolean;

Begin {Main Program}

```

```

Writeln('Begin Reading Fin.dft');

    Assign(Infile1,'a:\hreq.dat');
Reset(Infile1);
Assign(infile,'a:\fs.dat');
Reset(infile);
Assign(Outfile,'a:\fdat.dat');
Rewrite(Outfile);
Writeln('Reading Data');
repeat

Readln (Infile1,hfn,gts,bvn,evn,durn,tatn,fill);
    reset(infile);
    flg:=0;
    bvnh:=0;
    evnh:=10000;

    repeat

        readln (infile,hfn1,gts1,bvn1,evn1,durn1,tatn1,fill1);
        if gts=gts1 then

            begin
                bsn1:=bvn1-20;

                if (bvn>bsn1)and(bvn<evn1)and((durn+15)<(evn-evn1)) then bvn:=evn1+15;
                if (bvn>bsn1)and(bvn<evn1)and((durn+15)>(evn-evn1)) then flg:=1;
                if (evn>bsn1)and(evn<evn1)and(durn<(bsn1-bvn)) then evn:=bsn1;
                if (evn>bsn1)and(evn<evn1)and(durn>(bsn1-bvn)) then flg:=1;
                if (bvn1>bvn)and(evn1<evn)and((durn<(bsn1-bvn))or(durn<(evn-evn1))) then
                    begin
                        if (bsn1-bvn)>(evn-evn1-15) then evn:=bsn1;
                        if (bsn1-bvn)<(evn-evn1-15) then bvn:=evn1+15;
                    end;
                if (bvn1>bvn)and(evn1<evn)and((durn>(bsn1-bvn))and((durn+15)>(evn-evn1))) then flg:=1;
                if ((bvn-evn1)>0)and((bvn-evn1)<15) then bvn:=evn1+15;
                if bvn>bvnh then bvnh:=bvn;
                if evn<evnh then evnh:=evn;
            end;
        until eof(infile);
        if bvnh=0 then bvnh:=bvn;
        if evnh=10000 then evnh:=evn;
        if (evnh-bvnh)<durn then flg:=1;
    if flg=0 then writeln(outfile,hfn:4,gts,bvnh:5,evnh:5,durn:5,tatn:3,fill);
        bvnh:=0;
        evnh:=10000;

    until EOF (infile1);
Writeln('All done');
reset(outfile);
repeat

```

```
    readln(outfile);  
    until eof(outfile);  
  
end.
```

APPENDIX D

The Results from Random Sample Generation.

lreqrn.pas Output (Low altitude satellite support requests).

1 BOSS-A 101 151 50 20 747701111.1	21 COOK-B 1328 1344 16 20 458101111.1
1 BOSS-B 101 151 50 20 747701111.1	22 COOK-A 697 710 13 20 540901111.1
2 BOSS-A 565 577 12 20 780801111.1	22 COOK-B 697 710 13 20 540901111.1
2 BOSS-B 565 577 12 20 780801111.1	23 COOK-A 943 956 13 20 413901111.1
3 BOSS-A 316 321 5 20 757801111.1	23 COOK-B 943 956 13 20 413901111.1
3 BOSS-B 316 321 5 20 757801111.1	24 COOK-A 736 752 16 20 516001111.1
4 BOSS-A 1296 1341 45 20 891201111.1	24 COOK-B 736 752 16 20 516001111.1
4 BOSS-B 1296 1341 45 20 891201111.1	25 COOK-A 344 358 14 20 157601111.1
5 BOSS-A 210 219 9 20 611101111.1	25 COOK-B 344 358 14 20 157601111.1
5 BOSS-B 210 219 9 20 611101111.1	26 COOK-A 780 790 10 20 438301111.1
6 BOSS-A 689 704 15 20 725801111.1	26 COOK-B 780 790 10 20 438301111.1
6 BOSS-B 689 704 15 20 725801111.1	27 GUAM-A 40 50 10 20 380701111.1
7 BOSS-A 606 617 11 20 679101111.1	27 GUAM-B 40 50 10 20 380701111.1
7 BOSS-B 606 617 11 20 679101111.1	28 GUAM-A 999 1009 10 20 207501111.1
8 BOSS-A 548 588 40 20 636901111.1	28 GUAM-B 999 1009 10 20 207501111.1
8 BOSS-B 548 588 40 20 636901111.1	29 GUAM-A 1420 1432 12 20 782401111.1
9 BOSS-A 422 439 17 20 318101111.1	29 GUAM-B 1420 1432 12 20 782401111.1
9 BOSS-B 422 439 17 20 318101111.1	30 GUAM-A 1082 1094 12 20 389201111.1
10 BOSS-A 1135 1146 11 20 327001111.1	30 GUAM-B 1082 1094 12 20 389201111.1
10 BOSS-B 1135 1146 11 20 327001111.1	31 GUAM-A 1176 1189 13 20 644401111.1
11 BOSS-A 1416 1433 17 20 833301111.1	31 GUAM-B 1176 1189 13 20 644401111.1
11 BOSS-B 1416 1433 17 20 833301111.1	32 GUAM-A 846 863 17 20 337201111.1
12 BOSS-A 1166 1176 10 20 527601111.1	32 GUAM-B 846 863 17 20 337201111.1
12 BOSS-B 1166 1176 10 20 527601111.1	33 GUAM-A 1115 1127 12 20 188801111.1
13 BOSS-A 1304 1315 11 20 335501111.1	33 GUAM-B 1115 1127 12 20 188801111.1
13 BOSS-B 1304 1315 11 20 335501111.1	34 GUAM-A 288 302 14 20 899001111.1
14 BOSS-A 663 675 12 20 308601111.1	34 GUAM-B 288 302 14 20 899001111.1
14 BOSS-B 663 675 12 20 308601111.1	35 GUAM-A 33 44 11 20 120801111.1
15 BOSS-A 1365 1383 18 20 760401111.1	35 GUAM-B 33 44 11 20 120801111.1
15 BOSS-B 1365 1383 18 20 760401111.1	36 GUAM-A 631 643 12 20 637901111.1
16 BOSS-A 730 745 15 20 252401111.1	36 GUAM-B 631 643 12 20 637901111.1
16 BOSS-B 730 745 15 20 252401111.1	37 GUAM-A 464 481 17 20 110601111.1
17 BOSS-A 1076 1086 10 20 167301111.1	37 GUAM-B 464 481 17 20 110601111.1
17 BOSS-B 1076 1086 10 20 167301111.1	38 HULA-A 87 100 13 20 492101111.1
18 COOK-A 456 469 13 20 353501111.1	38 HULA-B 87 100 13 20 492101111.1
18 COOK-B 456 469 13 20 353501111.1	39 HULA-A 235 247 12 20 717701111.1
19 COOK-A 170 181 11 20 328801111.1	39 HULA-B 235 247 12 20 717701111.1
19 COOK-B 170 181 11 20 328801111.1	40 HULA-A 982 996 14 20 391501111.1
20 COOK-A 286 301 15 20 459901111.1	40 HULA-B 982 996 14 20 391501111.1
20 COOK-B 286 301 15 20 459901111.1	41 HULA-A 975 992 17 20 718301111.1
21 COOK-A 1328 1344 16 20 458101111.1	41 HULA-B 975 992 17 20 718301111.1

42 HULA-A 1238 1254	16 20 676201111.1	71 LION-A 1252 1266	14 20 398201111.1
42 HULA-B 1238 1254	16 20 676201111.1	71 LION-B 1252 1266	14 20 398201111.1
43 HULA-A 903 919	16 20 121101111.1	72 LION-A 704 718	14 20 232201111.1
43 HULA-B 903 919	16 20 121101111.1	72 LION-B 704 718	14 20 232201111.1
44 HULA-A 203 220	17 20 219301111.1	73 LION-A 1060 1073	13 20 285501111.1
44 HULA-B 203 220	17 20 219301111.1	73 LION-B 1060 1073	13 20 285501111.1
45 HULA-A 1018 1031	13 20 885101111.1	74 LION-A 1160 1174	14 20 666901111.1
45 HULA-B 1018 1031	13 20 885101111.1	74 LION-B 1160 1174	14 20 666901111.1
46 HULA-A 787 804	17 20 899101111.1	75 LION-A 1203 1216	13 20 741701111.1
46 HULA-B 787 804	17 20 899101111.1	75 LION-B 1203 1216	13 20 741701111.1
47 HULA-A 620 637	17 20 710901111.1	76 LION-A 1403 1419	16 20 276601111.1
47 HULA-B 620 637	17 20 710901111.1	76 LION-B 1403 1419	16 20 276601111.1
48 HULA-A 537 554	17 20 132401111.1	77 LION-A 616 629	13 20 239801111.1
48 HULA-B 537 554	17 20 132401111.1	77 LION-B 616 629	13 20 239801111.1
49 HULA-A 751 763	12 20 731801111.1	78 LION-A 346 358	12 20 772501111.1
49 HULA-B 751 763	12 20 731801111.1	78 LION-B 346 358	12 20 772501111.1
50 HULA-A 679 691	12 20 724101111.1	79 LION-A 1233 1247	14 20 580401111.1
50 HULA-B 679 691	12 20 724101111.1	79 LION-B 1233 1247	14 20 580401111.1
51 HULA-A 969 986	17 20 495401111.1	80 LION-A 280 292	12 20 874701111.1
51 HULA-B 969 986	17 20 495401111.1	80 LION-B 280 292	12 20 874701111.1
52 HULA-A 49 63	14 20 771401111.1	81 PIKE-A 1307 1323	16 20 163701111.1
52 HULA-B 49 63	14 20 771401111.1	82 PIKE-A 1032 1048	16 20 696201111.1
53 HULA-A 685 702	17 20 798901111.1	83 PIKE-A 651 667	16 20 897101111.1
53 HULA-B 685 702	17 20 798901111.1	84 PIKE-A 1078 1092	14 20 535201111.1
54 HULA-A 1031 1045	14 20 275501111.1	85 PIKE-A 948 961	13 20 533001111.1
54 HULA-B 1031 1045	14 20 275501111.1	86 PIKE-A 314 330	16 20 716501111.1
55 INDI-A 334 347	13 20 122701111.1	87 POGO-A 1384 1398	14 20 268601111.1
56 INDI-A 1136 1152	16 20 511401111.1	87 POGO-B 1384 1398	14 20 268601111.1
57 INDI-A 121 136	15 20 738801111.1	87 POGO-C 1384 1398	14 20 268601111.1
58 INDI-A 385 396	11 20 757101111.1	88 POGO-A 1224 1240	16 20 316201111.1
59 INDI-A 581 592	11 20 505101111.1	88 POGO-B 1224 1240	16 20 316201111.1
60 LION-A 103 118	15 20 252601111.1	88 POGO-C 1224 1240	16 20 316201111.1
60 LION-B 103 118	15 20 252601111.1	89 POGO-A 628 644	16 20 235101111.1
61 LION-A 1398 1414	16 20 409601111.1	89 POGO-B 628 644	16 20 235101111.1
61 LION-B 1398 1414	16 20 409601111.1	89 POGO-C 628 644	16 20 235101111.1
62 LION-A 771 785	14 20 758001111.1	90 POGO-A 777 791	14 20 594801111.1
62 LION-B 771 785	14 20 758001111.1	90 POGO-B 777 791	14 20 594801111.1
63 LION-A 942 956	14 20 748201111.1	90 POGO-C 777 791	14 20 594801111.1
63 LION-B 942 956	14 20 748201111.1	91 POGO-A 81 96	15 20 340501111.1
64 LION-A 1118 1132	14 20 591901111.1	91 POGO-B 81 96	15 20 340501111.1
64 LION-B 1118 1132	14 20 591901111.1	91 POGO-C 81 96	15 20 340501111.1
65 LION-A 968 981	13 20 794501111.1	92 POGO-A 761 776	15 20 404201111.1
65 LION-B 968 981	13 20 794501111.1	92 POGO-B 761 776	15 20 404201111.1
66 LION-A 1375 1388	13 20 517701111.1	92 POGO-C 761 776	15 20 404201111.1
66 LION-B 1375 1388	13 20 517701111.1	93 POGO-A 960 975	15 20 359301111.1
67 LION-A 996 1009	13 20 615801111.1	93 POGO-B 960 975	15 20 359301111.1
67 LION-B 996 1009	13 20 615801111.1	93 POGO-C 960 975	15 20 359301111.1
68 LION-A 1388 1401	13 20 601401111.1	94 POGO-A 460 474	14 20 283401111.1
68 LION-B 1388 1401	13 20 601401111.1	94 POGO-B 460 474	14 20 283401111.1
69 LION-A 1347 1359	12 20 220801111.1	94 POGO-C 460 474	14 20 283401111.1
69 LION-B 1347 1359	12 20 220801111.1	95 POGO-A 950 965	15 20 414401111.1
70 LION-A 1091 1105	14 20 880701111.1	95 POGO-B 950 965	15 20 414401111.1
70 LION-B 1091 1105	14 20 880701111.1	95 POGO-C 950 965	15 20 414401111.1

96 POGO-A	459 475	16 20 357701111.1	113 POGO-C	576 591	15 20 719101111.1
96 POGO-B	459 475	16 20 357701111.1	114 POGO-A	1298 1313	15 20 618201111.1
96 POGO-C	459 475	16 20 357701111.1	114 POGO-B	1298 1313	15 20 618201111.1
97 POGO-A	944 958	14 20 718001111.1	114 POGO-C	1298 1313	15 20 618201111.1
97 POGO-B	944 958	14 20 718001111.1	115 POGO-A	578 594	16 20 482901111.1
97 POGO-C	944 958	14 20 718001111.1	115 POGO-B	578 594	16 20 482901111.1
98 POGO-A	913 928	15 20 735501111.1	115 POGO-C	578 594	16 20 482901111.1
98 POGO-B	913 928	15 20 735501111.1	116 POGO-A	880 896	16 20 693901111.1
98 POGO-C	913 928	15 20 735501111.1	116 POGO-B	880 896	16 20 693901111.1
99 POGO-A	89 103	14 20 743201111.1	116 POGO-C	880 896	16 20 693901111.1
99 POGO-B	89 103	14 20 743201111.1	117 POGO-A	929 945	16 20 155301111.1
99 POGO-C	89 103	14 20 743201111.1	117 POGO-B	929 945	16 20 155301111.1
100 POGO-A	1309 1325	16 20 310401111.1	117 POGO-C	929 945	16 20 155301111.1
100 POGO-B	1309 1325	16 20 310401111.1	118 POGO-A	562 578	16 20 200001111.1
100 POGO-C	1309 1325	16 20 310401111.1	118 POGO-B	562 578	16 20 200001111.1
101 POGO-A	262 277	15 20 233201111.1	118 POGO-C	562 578	16 20 200001111.1
101 POGO-B	262 277	15 20 233201111.1	119 POGO-A	808 824	16 20 687101111.1
101 POGO-C	262 277	15 20 233201111.1	119 POGO-B	808 824	16 20 687101111.1
102 POGO-A	770 784	14 20 402401111.1	119 POGO-C	808 824	16 20 687101111.1
102 POGO-B	770 784	14 20 402401111.1	120 POGO-A	1285 1301	16 20 867701111.1
102 POGO-C	770 784	14 20 402401111.1	120 POGO-B	1285 1301	16 20 867701111.1
103 POGO-A	1267 1282	15 20 319501111.1	120 POGO-C	1285 1301	16 20 867701111.1
103 POGO-B	1267 1282	15 20 319501111.1	121 POGO-A	1236 1251	15 20 202801111.1
103 POGO-C	1267 1282	15 20 319501111.1	121 POGO-B	1236 1251	15 20 202801111.1
104 POGO-A	1133 1148	15 20 896801111.1	121 POGO-C	1236 1251	15 20 202801111.1
104 POGO-B	1133 1148	15 20 896801111.1	122 POGO-A	643 658	15 20 710301111.1
104 POGO-C	1133 1148	15 20 896801111.1	122 POGO-B	643 658	15 20 710301111.1
105 POGO-A	331 347	16 20 875401111.1	122 POGO-C	643 658	15 20 710301111.1
105 POGO-B	331 347	16 20 875401111.1	123 POGO-A	590 606	16 20 580501111.1
105 POGO-C	331 347	16 20 875401111.1	123 POGO-B	590 606	16 20 580501111.1
106 POGO-A	656 672	16 20 861701111.1	123 POGO-C	590 606	16 20 580501111.1
106 POGO-B	656 672	16 20 861701111.1	124 POGO-A	659 674	15 20 248501111.1
106 POGO-C	656 672	16 20 861701111.1	124 POGO-B	659 674	15 20 248501111.1
107 POGO-A	559 575	16 20 776201111.1	124 POGO-C	659 674	15 20 248501111.1
107 POGO-B	559 575	16 20 776201111.1	125 POGO-A	1218 1234	16 20 886901111.1
107 POGO-C	559 575	16 20 776201111.1	125 POGO-B	1218 1234	16 20 886901111.1
108 POGO-A	270 286	16 20 614101111.1	125 POGO-C	1218 1234	16 20 886901111.1
108 POGO-B	270 286	16 20 614101111.1	126 POGO-A	225 240	15 20 531201111.1
108 POGO-C	270 286	16 20 614101111.1	126 POGO-B	225 240	15 20 531201111.1
109 POGO-A	1366 1382	16 20 229801111.1	126 POGO-C	225 240	15 20 531201111.1
109 POGO-B	1366 1382	16 20 229801111.1	127 POGO-A	1012 1028	16 20 535601111.1
109 POGO-C	1366 1382	16 20 229801111.1	127 POGO-B	1012 1028	16 20 535601111.1
110 POGO-A	954 969	15 20 415901111.1	127 POGO-C	1012 1028	16 20 535601111.1
110 POGO-B	954 969	15 20 415901111.1	128 POGO-A	718 733	15 20 141801111.1
110 POGO-C	954 969	15 20 415901111.1	128 POGO-B	718 733	15 20 141801111.1
111 POGO-A	1137 1153	16 20 122901111.1	128 POGO-C	718 733	15 20 141801111.1
111 POGO-B	1137 1153	16 20 122901111.1	129 POGO-A	479 495	16 20 248801111.1
111 POGO-C	1137 1153	16 20 122901111.1	129 POGO-B	479 495	16 20 248801111.1
112 POGO-A	980 996	16 20 879301111.1	129 POGO-C	479 495	16 20 248801111.1
112 POGO-B	980 996	16 20 879301111.1	130 POGO-A	207 222	15 20 619401111.1
112 POGO-C	980 996	16 20 879301111.1	130 POGO-B	207 222	15 20 619401111.1
113 POGO-A	576 591	15 20 719101111.1	130 POGO-C	207 222	15 20 619401111.1
113 POGO-B	576 591	15 20 719101111.1	131 POGO-A	397 412	15 20 372501111.1

131 POGO-B 397 412 15 20 372501111.1
131 POGO-C 397 412 15 20 372501111.1
132 POGO-A 266 282 16 20 495001111.1
132 POGO-B 266 282 16 20 495001111.1
132 POGO-C 266 282 16 20 495001111.1
133 POGO-A 437 453 16 20 195701111.1
133 POGO-B 437 453 16 20 195701111.1
133 POGO-C 437 453 16 20 195701111.1
134 POGO-A 847 863 16 20 356501111.1
134 POGO-B 847 863 16 20 356501111.1
134 POGO-C 847 863 16 20 356501111.1
135 POGO-A 967 983 16 20 672801111.1

135 POGO-B 967 983 16 20 672801111.1
135 POGO-C 967 983 16 20 672801111.1
136 REEF-A 98 114 16 20 127501111.1
137 REEF-A 841 856 15 20 411301111.1
138 REEF-A 51 67 16 20 394201111.1
139 REEF-A 1321 1336 15 20 723501111.1
140 REEF-A 91 106 15 20 305201111.1
141 REEF-A 445 460 15 20 345001111.1
142 REEF-A 877 892 15 20 180601111.1
143 REEF-A 842 857 15 20 607001111.1
144 REEF-A 1156 1171 15 20 837501111.1
145 REEF-A 1381 1397 16 20 200101111.1

hreqrn.pas Output (Medium and High altitude satellite support requests).

1 POGO-A 1295 1345 15 15 7069	4 POGO-B 1365 1420 25 15 1225
1 POGO-B 1295 1345 15 15 7069	4 POGO-C 1365 1420 25 15 1225
1 POGO-C 1295 1345 15 15 7069	4 HULA-A 1365 1420 25 15 1225
1 HULA-A 1295 1345 15 15 7069	4 HULA-B 1365 1420 25 15 1225
1 HULA-B 1295 1345 15 15 7069	4 COOK-A 1365 1420 25 15 1225
1 COOK-A 1295 1345 15 15 7069	4 COOK-B 1365 1420 25 15 1225
1 COOK-B 1295 1345 15 15 7069	4 INDI-A 1365 1420 25 15 1225
1 INDI-A 1295 1345 15 15 7069	4 BOSS-A 1365 1420 25 15 1225
1 BOSS-A 1295 1345 15 15 7069	4 BOSS-B 1365 1420 25 15 1225
1 BOSS-B 1295 1345 15 15 7069	4 LION-A 1365 1420 25 15 1225
1 LION-A 1295 1345 15 15 7069	4 LION-B 1365 1420 25 15 1225
1 LION-B 1295 1345 15 15 7069	4 GUAM-A 1365 1420 25 15 1225
1 GUAM-A 1295 1345 15 15 7069	4 GUAM-B 1365 1420 25 15 1225
1 GUAM-B 1295 1345 15 15 7069	4 PIKE-A 1365 1420 25 15 1225
1 PIKE-A 1295 1345 15 15 7069	4 REEF-A 1365 1420 25 15 1225
1 REEF-A 1295 1345 15 15 7069	5 POGO-A 370 425 35 15 6729
2 POGO-A 80 515 435 15 6568	5 POGO-B 370 425 35 15 6729
2 POGO-B 80 515 435 15 6568	5 POGO-C 370 425 35 15 6729
2 POGO-C 80 515 435 15 6568	5 HULA-A 370 425 35 15 6729
2 HULA-A 80 515 435 15 6568	5 HULA-B 370 425 35 15 6729
2 HULA-B 80 515 435 15 6568	5 COOK-A 370 425 35 15 6729
2 COOK-A 80 515 435 15 6568	5 COOK-B 370 425 35 15 6729
2 COOK-B 80 515 435 15 6568	5 INDI-A 370 425 35 15 6729
2 INDI-A 80 515 435 15 6568	5 BOSS-A 370 425 35 15 6729
2 BOSS-A 80 515 435 15 6568	5 BOSS-B 370 425 35 15 6729
2 BOSS-B 80 515 435 15 6568	5 LION-A 370 425 35 15 6729
2 LION-A 80 515 435 15 6568	5 LION-B 370 425 35 15 6729
2 LION-B 80 515 435 15 6568	5 GUAM-A 370 425 35 15 6729
2 GUAM-A 80 515 435 15 6568	5 GUAM-B 370 425 35 15 6729
2 GUAM-B 80 515 435 15 6568	5 PIKE-A 370 425 35 15 6729
2 PIKE-A 80 515 435 15 6568	5 REEF-A 370 425 35 15 6729
2 REEF-A 80 515 435 15 6568	6 POGO-A 1030 1080 50 15 3274
3 POGO-A 365 445 20 15 5283	6 POGO-B 1030 1080 50 15 3274
3 POGO-B 365 445 20 15 5283	6 POGO-C 1030 1080 50 15 3274
3 POGO-C 365 445 20 15 5283	6 HULA-A 1030 1080 50 15 3274
3 HULA-A 365 445 20 15 5283	6 HULA-B 1030 1080 50 15 3274
3 HULA-B 365 445 20 15 5283	6 COOK-A 1030 1080 50 15 3274
3 COOK-A 365 445 20 15 5283	6 COOK-B 1030 1080 50 15 3274
3 COOK-B 365 445 20 15 5283	6 INDI-A 1030 1080 50 15 3274
3 INDI-A 365 445 20 15 5283	6 BOSS-A 1030 1080 50 15 3274
3 BOSS-A 365 445 20 15 5283	6 BOSS-B 1030 1080 50 15 3274
3 BOSS-B 365 445 20 15 5283	6 LION-A 1030 1080 50 15 3274
3 LION-A 365 445 20 15 5283	6 LION-B 1030 1080 50 15 3274
3 LION-B 365 445 20 15 5283	6 GUAM-A 1030 1080 50 15 3274
3 GUAM-A 365 445 20 15 5283	6 GUAM-B 1030 1080 50 15 3274
3 GUAM-B 365 445 20 15 5283	6 PIKE-A 1030 1080 50 15 3274
3 PIKE-A 365 445 20 15 5283	6 REEF-A 1030 1080 50 15 3274
3 REEF-A 365 445 20 15 5283	7 POGO-A 100 150 40 15 7780
4 POGO-A 1365 1420 25 15 1225	7 POGO-B 100 150 40 15 7780

7 POGO-C 100 150 40 15 7780
 7 HULA-A 100 150 40 15 7780
 7 HULA-B 100 150 40 15 7780
 7 COOK-A 100 150 40 15 7780
 7 COOK-B 100 150 40 15 7780
 7 INDI-A 100 150 40 15 7780
 7 BOSS-A 100 150 40 15 7780
 7 BOSS-B 100 150 40 15 7780
 7 LION-A 100 150 40 15 7780
 7 LION-B 100 150 40 15 7780
 7 GUAM-A 100 150 40 15 7780
 7 GUAM-B 100 150 40 15 7780
 7 PIKE-A 100 150 40 15 7780
 7 REEF-A 100 150 40 15 7780
 8 POGO-A 935 985 50 15 5724
 8 POGO-B 935 985 50 15 5724
 8 POGO-C 935 985 50 15 5724
 8 HULA-A 935 985 50 15 5724
 8 HULA-B 935 985 50 15 5724
 8 COOK-A 935 985 50 15 5724
 8 COOK-B 935 985 50 15 5724
 8 INDI-A 935 985 50 15 5724
 8 BOSS-A 935 985 50 15 5724
 8 BOSS-B 935 985 50 15 5724
 8 LION-A 935 985 50 15 5724
 8 LION-B 935 985 50 15 5724
 8 GUAM-A 935 985 50 15 5724
 8 GUAM-B 935 985 50 15 5724
 8 PIKE-A 935 985 50 15 5724
 8 REEF-A 935 985 50 15 5724
 9 POGO-A 285 335 45 15 8928
 9 POGO-B 285 335 45 15 8928
 9 POGO-C 285 335 45 15 8928
 9 HULA-A 285 335 45 15 8928
 9 HULA-B 285 335 45 15 8928
 9 COOK-A 285 335 45 15 8928
 9 COOK-B 285 335 45 15 8928
 9 INDI-A 285 335 45 15 8928
 9 BOSS-A 285 335 45 15 8928
 9 BOSS-B 285 335 45 15 8928
 9 LION-A 285 335 45 15 8928
 9 LION-B 285 335 45 15 8928
 9 GUAM-A 285 335 45 15 8928
 9 GUAM-B 285 335 45 15 8928
 9 PIKE-A 285 335 45 15 8928
 9 REEF-A 285 335 45 15 8928
 10 POGO-A 760 810 45 15 2937
 10 POGO-B 760 810 45 15 2937
 10 POGO-C 760 810 45 15 2937
 10 HULA-A 760 810 45 15 2937
 10 HULA-B 760 810 45 15 2937
 10 COOK-A 760 810 45 15 2937
 10 COOK-B 760 810 45 15 2937

10 INDI-A 760 810 45 15 2937
 10 BOSS-A 760 810 45 15 2937
 10 BOSS-B 760 810 45 15 2937
 10 LION-A 760 810 45 15 2937
 10 LION-B 760 810 45 15 2937
 10 GUAM-A 760 810 45 15 2937
 10 GUAM-B 760 810 45 15 2937
 10 PIKE-A 760 810 45 15 2937
 10 REEF-A 760 810 45 15 2937
 11 POGO-A 255 305 40 15 2381
 11 POGO-B 255 305 40 15 2381
 11 POGO-C 255 305 40 15 2381
 11 HULA-A 255 305 40 15 2381
 11 HULA-B 255 305 40 15 2381
 11 COOK-A 255 305 40 15 2381
 11 COOK-B 255 305 40 15 2381
 11 INDI-A 255 305 40 15 2381
 11 BOSS-A 255 305 40 15 2381
 11 BOSS-B 255 305 40 15 2381
 11 LION-A 255 305 40 15 2381
 11 LION-B 255 305 40 15 2381
 11 GUAM-A 255 305 40 15 2381
 11 GUAM-B 255 305 40 15 2381
 11 PIKE-A 255 305 40 15 2381
 11 REEF-A 255 305 40 15 2381
 12 POGO-A 235 285 10 15 6618
 12 POGO-B 235 285 10 15 6618
 12 POGO-C 235 285 10 15 6618
 12 HULA-A 235 285 10 15 6618
 12 HULA-B 235 285 10 15 6618
 12 COOK-A 235 285 10 15 6618
 12 COOK-B 235 285 10 15 6618
 12 INDI-A 235 285 10 15 6618
 12 BOSS-A 235 285 10 15 6618
 12 BOSS-B 235 285 10 15 6618
 12 LION-A 235 285 10 15 6618
 12 LION-B 235 285 10 15 6618
 12 GUAM-A 235 285 10 15 6618
 12 GUAM-B 235 285 10 15 6618
 12 PIKE-A 235 285 10 15 6618
 12 REEF-A 235 285 10 15 6618
 13 POGO-A 340 640 300 15 4848
 13 POGO-B 340 640 300 15 4848
 13 POGO-C 340 640 300 15 4848
 13 HULA-A 340 640 300 15 4848
 13 HULA-B 340 640 300 15 4848
 13 COOK-A 340 640 300 15 4848
 13 COOK-B 340 640 300 15 4848
 13 INDI-A 340 640 300 15 4848
 13 BOSS-A 340 640 300 15 4848
 13 BOSS-B 340 640 300 15 4848
 13 LION-A 340 640 300 15 4848
 13 LION-B 340 640 300 15 4848

13 GUAM-A 340 640 300 15 4848
 13 GUAM-B 340 640 300 15 4848
 13 PIKE-A 340 640 300 15 4848
 13 REEF-A 340 640 300 15 4848
 14 POGO-A 1275 1440 165 15 1702
 14 POGO-B 1275 1440 165 15 1702
 14 POGO-C 1275 1440 165 15 1702
 14 HULA-A 1275 1440 165 15 1702
 14 HULA-B 1275 1440 165 15 1702
 14 COOK-A 1275 1440 165 15 1702
 14 COOK-B 1275 1440 165 15 1702
 14 INDI-A 1275 1440 165 15 1702
 14 BOSS-A 1275 1440 165 15 1702
 14 BOSS-B 1275 1440 165 15 1702
 14 LION-A 1275 1440 165 15 1702
 14 LION-B 1275 1440 165 15 1702
 14 GUAM-A 1275 1440 165 15 1702
 14 GUAM-B 1275 1440 165 15 1702
 14 PIKE-A 1275 1440 165 15 1702
 14 REEF-A 1275 1440 165 15 1702
 15 POGO-A 1245 1300 35 15 6809
 15 POGO-B 1245 1300 35 15 6809
 15 POGO-C 1245 1300 35 15 6809
 15 HULA-A 1245 1300 35 15 6809
 15 HULA-B 1245 1300 35 15 6809
 15 COOK-A 1245 1300 35 15 6809
 15 COOK-B 1245 1300 35 15 6809
 15 INDI-A 1245 1300 35 15 6809
 15 BOSS-A 1245 1300 35 15 6809
 15 BOSS-B 1245 1300 35 15 6809
 15 LION-A 1245 1300 35 15 6809
 15 LION-B 1245 1300 35 15 6809
 15 GUAM-A 1245 1300 35 15 6809
 15 GUAM-B 1245 1300 35 15 6809
 15 PIKE-A 1245 1300 35 15 6809
 15 REEF-A 1245 1300 35 15 6809
 16 POGO-A 45 85 30 15 8824
 16 POGO-B 45 85 30 15 8824
 16 POGO-C 45 85 30 15 8824
 16 HULA-A 45 85 30 15 8824
 16 HULA-B 45 85 30 15 8824
 16 COOK-A 45 85 30 15 8824
 16 COOK-B 45 85 30 15 8824
 16 INDI-A 45 85 30 15 8824
 16 BOSS-A 45 85 30 15 8824
 16 BOSS-B 45 85 30 15 8824
 16 LION-A 45 85 30 15 8824
 16 LION-B 45 85 30 15 8824
 16 GUAM-A 45 85 30 15 8824
 16 GUAM-B 45 85 30 15 8824
 16 PIKE-A 45 85 30 15 8824
 16 REEF-A 45 85 30 15 8824
 17 POGO-A 645 695 10 15 2848

17 POGO-B 645 695 10 15 2848
 17 POGO-C 645 695 10 15 2848
 17 HULA-A 645 695 10 15 2848
 17 HULA-B 645 695 10 15 2848
 17 COOK-A 645 695 10 15 2848
 17 COOK-B 645 695 10 15 2848
 17 INDI-A 645 695 10 15 2848
 17 BOSS-A 645 695 10 15 2848
 17 BOSS-B 645 695 10 15 2848
 17 LION-A 645 695 10 15 2848
 17 LION-B 645 695 10 15 2848
 17 GUAM-A 645 695 10 15 2848
 17 GUAM-B 645 695 10 15 2848
 17 PIKE-A 645 695 10 15 2848
 17 REEF-A 645 695 10 15 2848
 18 INDI-A 755 805 15 15 2116
 18 LION-A 755 805 15 15 2116
 18 LION-B 755 805 15 15 2116
 18 REEF-A 755 805 15 15 2116
 19 INDI-A 875 925 10 15 8569
 19 LION-A 875 925 10 15 8569
 19 LION-B 875 925 10 15 8569
 19 REEF-A 875 925 10 15 8569
 20 INDI-A 600 680 20 15 1958
 20 LION-A 600 680 20 15 1958
 20 LION-B 600 680 20 15 1958
 20 REEF-A 600 680 20 15 1958
 21 INDI-A 925 1005 20 15 5187
 21 LION-A 925 1005 20 15 5187
 21 LION-B 925 1005 20 15 5187
 21 REEF-A 925 1005 20 15 5187
 22 INDI-A 690 740 50 15 4849
 22 LION-A 690 740 50 15 4849
 22 LION-B 690 740 50 15 4849
 22 REEF-A 690 740 50 15 4849
 23 INDI-A 670 720 40 15 1317
 23 LION-A 670 720 40 15 1317
 23 LION-B 670 720 40 15 1317
 23 REEF-A 670 720 40 15 1317
 24 INDI-A 225 265 30 15 1774
 24 LION-A 225 265 30 15 1774
 24 LION-B 225 265 30 15 1774
 24 REEF-A 225 265 30 15 1774
 25 INDI-A 1330 1380 45 15 3317
 25 LION-A 1330 1380 45 15 3317
 25 LION-B 1330 1380 45 15 3317
 25 REEF-A 1330 1380 45 15 3317
 26 INDI-A 400 450 15 15 5656
 26 LION-A 400 450 15 15 5656
 26 LION-B 400 450 15 15 5656
 26 REEF-A 400 450 15 15 5656
 27 INDI-A 1160 1240 20 15 7107
 27 LION-A 1160 1240 20 15 7107

27 LION-B 1160 1240 20 15 7107
 27 REEF-A 1160 1240 20 15 7107
 28 INDI-A 335 375 30 15 8579
 28 LION-A 335 375 30 15 8579
 28 LION-B 335 375 30 15 8579
 28 REEF-A 335 375 30 15 8579
 29 INDI-A 775 825 10 15 8414
 29 LION-A 775 825 10 15 8414
 29 LION-B 775 825 10 15 8414
 29 REEF-A 775 825 10 15 8414
 30 INDI-A 1035 1085 15 15 4169
 30 LION-A 1035 1085 15 15 4169
 30 LION-B 1035 1085 15 15 4169
 30 REEF-A 1035 1085 15 15 4169
 31 INDI-A 185 240 35 15 1002
 31 LION-A 185 240 35 15 1002
 31 LION-B 185 240 35 15 1002
 31 REEF-A 185 240 35 15 1002
 32 INDI-A 970 1050 20 15 7492
 32 LION-A 970 1050 20 15 7492
 32 LION-B 970 1050 20 15 7492
 32 REEF-A 970 1050 20 15 7492
 33 INDI-A 1220 1270 15 15 3149
 33 BOSS-A 1220 1270 15 15 3149
 33 BOSS-B 1220 1270 15 15 3149
 33 LION-A 1220 1270 15 15 3149
 33 LION-B 1220 1270 15 15 3149
 34 INDI-A 55 95 30 15 2118
 34 BOSS-A 55 95 30 15 2118
 34 BOSS-B 55 95 30 15 2118
 34 LION-A 55 95 30 15 2118
 34 LION-B 55 95 30 15 2118
 35 INDI-A 155 205 45 15 6102
 35 BOSS-A 155 205 45 15 6102
 35 BOSS-B 155 205 45 15 6102
 35 LION-A 155 205 45 15 6102
 35 LION-B 155 205 45 15 6102
 36 INDI-A 265 315 15 15 6755
 36 BOSS-A 265 315 15 15 6755
 36 BOSS-B 265 315 15 15 6755
 36 LION-A 265 315 15 15 6755
 36 LION-B 265 315 15 15 6755
 37 INDI-A 1010 1060 10 15 7213
 37 BOSS-A 1010 1060 10 15 7213
 37 BOSS-B 1010 1060 10 15 7213
 37 LION-A 1010 1060 10 15 7213
 37 LION-B 1010 1060 10 15 7213
 38 INDI-A 140 195 35 15 6928
 38 BOSS-A 140 195 35 15 6928
 38 BOSS-B 140 195 35 15 6928
 38 LION-A 140 195 35 15 6928
 38 LION-B 140 195 35 15 6928
 39 INDI-A 1005 1070 5 15 6659

39 BOSS-A 1005 1070 5 15 6659
 39 BOSS-B 1005 1070 5 15 6659
 39 LION-A 1005 1070 5 15 6659
 39 LION-B 1005 1070 5 15 6659
 40 INDI-A 735 785 45 15 8031
 40 BOSS-A 735 785 45 15 8031
 40 BOSS-B 735 785 45 15 8031
 40 LION-A 735 785 45 15 8031
 40 LION-B 735 785 45 15 8031
 41 INDI-A 975 1055 20 15 3061
 41 BOSS-A 975 1055 20 15 3061
 41 BOSS-B 975 1055 20 15 3061
 41 LION-A 975 1055 20 15 3061
 41 LION-B 975 1055 20 15 3061
 42 INDI-A 115 170 35 15 3833
 42 BOSS-A 115 170 35 15 3833
 42 BOSS-B 115 170 35 15 3833
 42 LION-A 115 170 35 15 3833
 42 LION-B 115 170 35 15 3833
 43 INDI-A 845 925 20 15 4947
 43 BOSS-A 845 925 20 15 4947
 43 BOSS-B 845 925 20 15 4947
 43 LION-A 845 925 20 15 4947
 43 LION-B 845 925 20 15 4947
 44 INDI-A 305 360 25 15 8183
 44 BOSS-A 305 360 25 15 8183
 44 BOSS-B 305 360 25 15 8183
 44 LION-A 305 360 25 15 8183
 44 LION-B 305 360 25 15 8183
 45 INDI-A 1320 1370 10 15 8971
 45 BOSS-A 1320 1370 10 15 8971
 45 BOSS-B 1320 1370 10 15 8971
 45 LION-A 1320 1370 10 15 8971
 45 LION-B 1320 1370 10 15 8971
 46 INDI-A 1345 1400 35 15 8988
 46 BOSS-A 1345 1400 35 15 8988
 46 BOSS-B 1345 1400 35 15 8988
 46 LION-A 1345 1400 35 15 8988
 46 LION-B 1345 1400 35 15 8988
 47 BOSS-A 650 700 15 15 6355
 47 BOSS-B 650 700 15 15 6355
 47 LION-A 650 700 15 15 6355
 47 LION-B 650 700 15 15 6355
 48 BOSS-A 415 495 20 15 2190
 48 BOSS-B 415 495 20 15 2190
 48 LION-A 415 495 20 15 2190
 48 LION-B 415 495 20 15 2190
 49 BOSS-A 1240 1320 20 15 3242
 49 BOSS-B 1240 1320 20 15 3242
 49 LION-A 1240 1320 20 15 3242
 49 LION-B 1240 1320 20 15 3242
 50 BOSS-A 1075 1125 45 15 5522
 50 BOSS-B 1075 1125 45 15 5522

50 LION-A 1075 1125 45 15 5522
 50 LION-B 1075 1125 45 15 5522
 51 BOSS-A 1185 1265 20 15 8492
 51 BOSS-B 1185 1265 20 15 8492
 51 LION-A 1185 1265 20 15 8492
 51 LION-B 1185 1265 20 15 8492
 52 BOSS-A 1340 1390 45 15 8983
 52 BOSS-B 1340 1390 45 15 8983
 52 LION-A 1340 1390 45 15 8983
 52 LION-B 1340 1390 45 15 8983
 53 BOSS-A 980 1030 15 15 6230
 53 BOSS-B 980 1030 15 15 6230
 53 LION-A 980 1030 15 15 6230
 53 LION-B 980 1030 15 15 6230
 54 BOSS-A 200 250 40 15 6364
 54 BOSS-B 200 250 40 15 6364
 54 LION-A 200 250 40 15 6364
 54 LION-B 200 250 40 15 6364
 55 BOSS-A 15 65 15 15 3800
 55 BOSS-B 15 65 15 15 3800
 55 LION-A 15 65 15 15 3800
 55 LION-B 15 65 15 15 3800
 56 BOSS-A 1390 1440 45 15 2957
 56 BOSS-B 1390 1440 45 15 2957
 56 LION-A 1390 1440 45 15 2957
 56 LION-B 1390 1440 45 15 2957
 57 BOSS-A 685 740 25 15 8977
 57 BOSS-B 685 740 25 15 8977
 57 LION-A 685 740 25 15 8977
 57 LION-B 685 740 25 15 8977
 58 BOSS-A 950 1000 15 15 6617
 58 BOSS-B 950 1000 15 15 6617
 58 LION-A 950 1000 15 15 6617
 58 LION-B 950 1000 15 15 6617
 59 BOSS-A 345 395 15 15 8649
 59 BOSS-B 345 395 15 15 8649
 59 LION-A 345 395 15 15 8649
 59 LION-B 345 395 15 15 8649
 60 HULA-A 810 860 10 15 2569
 60 HULA-B 810 860 10 15 2569
 60 COOK-A 810 860 10 15 2569
 60 COOK-B 810 860 10 15 2569
 60 BOSS-A 810 860 10 15 2569
 60 BOSS-B 810 860 10 15 2569
 60 PIKE-A 810 860 10 15 2569
 61 HULA-A 90 170 20 15 2110
 61 HULA-B 90 170 20 15 2110
 61 COOK-A 90 170 20 15 2110
 61 COOK-B 90 170 20 15 2110
 61 BOSS-A 90 170 20 15 2110
 61 BOSS-B 90 170 20 15 2110
 61 PIKE-A 90 170 20 15 2110
 62 HULA-A 990 1040 40 15 6938

62 HULA-B 990 1040 40 15 6938
 62 COOK-A 990 1040 40 15 6938
 62 COOK-B 990 1040 40 15 6938
 62 BOSS-A 990 1040 40 15 6938
 62 BOSS-B 990 1040 40 15 6938
 62 PIKE-A 990 1040 40 15 6938
 63 HULA-A 1235 1300 5 15 7612
 63 HULA-B 1235 1300 5 15 7612
 63 COOK-A 1235 1300 5 15 7612
 63 COOK-B 1235 1300 5 15 7612
 63 BOSS-A 1235 1300 5 15 7612
 63 BOSS-B 1235 1300 5 15 7612
 63 PIKE-A 1235 1300 5 15 7612
 64 HULA-A 360 660 300 15 8438
 64 HULA-B 360 660 300 15 8438
 64 COOK-A 360 660 300 15 8438
 64 COOK-B 360 660 300 15 8438
 64 BOSS-A 360 660 300 15 8438
 64 BOSS-B 360 660 300 15 8438
 64 PIKE-A 360 660 300 15 8438
 65 HULA-A 450 500 15 15 4781
 65 HULA-B 450 500 15 15 4781
 65 COOK-A 450 500 15 15 4781
 65 COOK-B 450 500 15 15 4781
 65 BOSS-A 450 500 15 15 4781
 65 BOSS-B 450 500 15 15 4781
 65 PIKE-A 450 500 15 15 4781
 66 HULA-A 1280 1330 15 15 4638
 66 HULA-B 1280 1330 15 15 4638
 66 COOK-A 1280 1330 15 15 4638
 66 COOK-B 1280 1330 15 15 4638
 66 BOSS-A 1280 1330 15 15 4638
 66 BOSS-B 1280 1330 15 15 4638
 66 PIKE-A 1280 1330 15 15 4638
 67 HULA-A 245 295 15 15 1608
 67 HULA-B 245 295 15 15 1608
 67 COOK-A 245 295 15 15 1608
 67 COOK-B 245 295 15 15 1608
 67 BOSS-A 245 295 15 15 1608
 67 BOSS-B 245 295 15 15 1608
 67 PIKE-A 245 295 15 15 1608
 68 HULA-A 170 220 10 15 7986
 68 HULA-B 170 220 10 15 7986
 68 COOK-A 170 220 10 15 7986
 68 COOK-B 170 220 10 15 7986
 68 BOSS-A 170 220 10 15 7986
 68 BOSS-B 170 220 10 15 7986
 68 PIKE-A 170 220 10 15 7986
 69 HULA-A 1150 1215 5 15 2541
 69 HULA-B 1150 1215 5 15 2541
 69 COOK-A 1150 1215 5 15 2541
 69 COOK-B 1150 1215 5 15 2541
 69 BOSS-A 1150 1215 5 15 2541

69 BOSS-B 1150 1215 5 15 2541
 69 PIKE-A 1150 1215 5 15 2541
 70 HULA-A 495 545 15 15 3599
 70 HULA-B 495 545 15 15 3599
 70 COOK-A 495 545 15 15 3599
 70 COOK-B 495 545 15 15 3599
 70 BOSS-A 495 545 15 15 3599
 70 BOSS-B 495 545 15 15 3599
 70 PIKE-A 495 545 15 15 3599
 71 HULA-A 30 85 25 15 8369
 71 HULA-B 30 85 25 15 8369
 71 COOK-A 30 85 25 15 8369
 71 COOK-B 30 85 25 15 8369
 71 BOSS-A 30 85 25 15 8369
 71 BOSS-B 30 85 25 15 8369
 71 PIKE-A 30 85 25 15 8369
 72 BOSS-A 665 715 10 15 4345
 72 BOSS-B 665 715 10 15 4345
 72 LION-A 665 715 10 15 4345
 72 LION-B 665 715 10 15 4345
 72 PIKE-A 665 715 10 15 4345
 73 BOSS-A 50 115 5 15 4483
 73 BOSS-B 50 115 5 15 4483
 73 LION-A 50 115 5 15 4483
 73 LION-B 50 115 5 15 4483
 73 PIKE-A 50 115 5 15 4483
 74 BOSS-A 825 875 15 15 2463
 74 BOSS-B 825 875 15 15 2463
 74 LION-A 825 875 15 15 2463
 74 LION-B 825 875 15 15 2463
 74 PIKE-A 825 875 15 15 2463
 75 BOSS-A 520 600 20 15 2618
 75 BOSS-B 520 600 20 15 2618
 75 LION-A 520 600 20 15 2618
 75 LION-B 520 600 20 15 2618
 75 PIKE-A 520 600 20 15 2618
 76 BOSS-A 355 410 25 15 6853
 76 BOSS-B 355 410 25 15 6853
 76 LION-A 355 410 25 15 6853
 76 LION-B 355 410 25 15 6853
 76 PIKE-A 355 410 25 15 6853
 77 BOSS-A 515 580 5 15 7979
 77 BOSS-B 515 580 5 15 7979
 77 LION-A 515 580 5 15 7979
 77 LION-B 515 580 5 15 7979
 77 PIKE-A 515 580 5 15 7979
 78 BOSS-A 270 320 15 15 6115
 78 BOSS-B 270 320 15 15 6115
 78 LION-A 270 320 15 15 6115
 78 LION-B 270 320 15 15 6115
 78 PIKE-A 270 320 15 15 6115
 79 BOSS-A 1115 1165 15 15 5886
 79 BOSS-B 1115 1165 15 15 5886

79 LION-A 1115 1165 15 15 5886
 79 LION-B 1115 1165 15 15 5886
 79 PIKE-A 1115 1165 15 15 5886
 80 BOSS-A 795 845 15 15 4978
 80 BOSS-B 795 845 15 15 4978
 80 LION-A 795 845 15 15 4978
 80 LION-B 795 845 15 15 4978
 80 PIKE-A 795 845 15 15 4978
 81 BOSS-A 955 1005 10 15 4822
 81 BOSS-B 955 1005 10 15 4822
 81 LION-A 955 1005 10 15 4822
 81 LION-B 955 1005 10 15 4822
 81 PIKE-A 955 1005 10 15 4822
 82 BOSS-A 1195 1275 20 15 4374
 82 BOSS-B 1195 1275 20 15 4374
 82 LION-A 1195 1275 20 15 4374
 82 LION-B 1195 1275 20 15 4374
 82 PIKE-A 1195 1275 20 15 4374
 83 BOSS-A 1105 1185 20 15 5226
 83 BOSS-B 1105 1185 20 15 5226
 83 LION-A 1105 1185 20 15 5226
 83 LION-B 1105 1185 20 15 5226
 83 PIKE-A 1105 1185 20 15 5226
 84 INDI-A 1080 1135 25 15 1614
 84 LION-A 1080 1135 25 15 1614
 84 LION-B 1080 1135 25 15 1614
 84 GUAM-A 1080 1135 25 15 1614
 84 GUAM-B 1080 1135 25 15 1614
 84 REEF-A 1080 1135 25 15 1614
 85 INDI-A 1190 1240 15 15 5033
 85 LION-A 1190 1240 15 15 5033
 85 LION-B 1190 1240 15 15 5033
 85 GUAM-A 1190 1240 15 15 5033
 85 GUAM-B 1190 1240 15 15 5033
 85 REEF-A 1190 1240 15 15 5033
 86 INDI-A 595 650 25 15 7105
 86 LION-A 595 650 25 15 7105
 86 LION-B 595 650 25 15 7105
 86 GUAM-A 595 650 25 15 7105
 86 GUAM-B 595 650 25 15 7105
 86 REEF-A 595 650 25 15 7105
 87 INDI-A 1300 1350 15 15 8858
 87 LION-A 1300 1350 15 15 8858
 87 LION-B 1300 1350 15 15 8858
 87 GUAM-A 1300 1350 15 15 8858
 87 GUAM-B 1300 1350 15 15 8858
 87 REEF-A 1300 1350 15 15 8858
 88 INDI-A 1140 1190 15 15 3192
 88 LION-A 1140 1190 15 15 3192
 88 LION-B 1140 1190 15 15 3192
 88 GUAM-A 1140 1190 15 15 3192
 88 GUAM-B 1140 1190 15 15 3192
 88 REEF-A 1140 1190 15 15 3192

89 INDI-A 635 715 20 15 7686
 89 LION-A 635 715 20 15 7686
 89 LION-B 635 715 20 15 7686
 89 GUAM-A 635 715 20 15 7686
 89 GUAM-B 635 715 20 15 7686
 89 REEF-A 635 715 20 15 7686
 90 INDI-A 410 490 20 15 7831
 90 LION-A 410 490 20 15 7831
 90 LION-B 410 490 20 15 7831
 90 GUAM-A 410 490 20 15 7831
 90 GUAM-B 410 490 20 15 7831
 90 REEF-A 410 490 20 15 7831
 91 INDI-A 230 280 10 15 8778
 91 LION-A 230 280 10 15 8778
 91 LION-B 230 280 10 15 8778
 91 GUAM-A 230 280 10 15 8778
 91 GUAM-B 230 280 10 15 8778
 91 REEF-A 230 280 10 15 8778
 92 INDI-A 540 590 15 15 8157
 92 LION-A 540 590 15 15 8157
 92 LION-B 540 590 15 15 8157
 92 GUAM-A 540 590 15 15 8157
 92 GUAM-B 540 590 15 15 8157
 92 REEF-A 540 590 15 15 8157
 93 INDI-A 375 425 10 15 7331
 93 LION-A 375 425 10 15 7331
 93 LION-B 375 425 10 15 7331
 93 GUAM-A 375 425 10 15 7331
 93 GUAM-B 375 425 10 15 7331
 93 REEF-A 375 425 10 15 7331
 94 INDI-A 1350 1430 20 15 4284
 94 LION-A 1350 1430 20 15 4284
 94 LION-B 1350 1430 20 15 4284
 94 GUAM-A 1350 1430 20 15 4284
 94 GUAM-B 1350 1430 20 15 4284
 94 REEF-A 1350 1430 20 15 4284
 95 POGO-A 295 360 5 15 5218
 95 POGO-B 295 360 5 15 5218
 95 POGO-C 295 360 5 15 5218
 95 HULA-A 295 360 5 15 5218
 95 HULA-B 295 360 5 15 5218
 95 COOK-A 295 360 5 15 5218
 95 COOK-B 295 360 5 15 5218
 95 BOSS-A 295 360 5 15 5218
 95 BOSS-B 295 360 5 15 5218
 95 LION-A 295 360 5 15 5218
 95 LION-B 295 360 5 15 5218
 95 GUAM-A 295 360 5 15 5218
 95 GUAM-B 295 360 5 15 5218
 95 PIKE-A 295 360 5 15 5218
 96 POGO-A 20 100 20 15 4117
 96 POGO-B 20 100 20 15 4117
 96 POGO-C 20 100 20 15 4117

96 HULA-A 20 100 20 15 4117
 96 HULA-B 20 100 20 15 4117
 96 COOK-A 20 100 20 15 4117
 96 COOK-B 20 100 20 15 4117
 96 BOSS-A 20 100 20 15 4117
 96 BOSS-B 20 100 20 15 4117
 96 LION-A 20 100 20 15 4117
 96 LION-B 20 100 20 15 4117
 96 GUAM-A 20 100 20 15 4117
 96 GUAM-B 20 100 20 15 4117
 96 PIKE-A 20 100 20 15 4117
 97 POGO-A 10 75 5 15 8308
 97 POGO-B 10 75 5 15 8308
 97 POGO-C 10 75 5 15 8308
 97 HULA-A 10 75 5 15 8308
 97 HULA-B 10 75 5 15 8308
 97 COOK-A 10 75 5 15 8308
 97 COOK-B 10 75 5 15 8308
 97 BOSS-A 10 75 5 15 8308
 97 BOSS-B 10 75 5 15 8308
 97 LION-A 10 75 5 15 8308
 97 LION-B 10 75 5 15 8308
 97 GUAM-A 10 75 5 15 8308
 97 GUAM-B 10 75 5 15 8308
 97 PIKE-A 10 75 5 15 8308
 98 POGO-A 440 490 15 15 3481
 98 POGO-B 440 490 15 15 3481
 98 POGO-C 440 490 15 15 3481
 98 HULA-A 440 490 15 15 3481
 98 HULA-B 440 490 15 15 3481
 98 COOK-A 440 490 15 15 3481
 98 COOK-B 440 490 15 15 3481
 98 BOSS-A 440 490 15 15 3481
 98 BOSS-B 440 490 15 15 3481
 98 LION-A 440 490 15 15 3481
 98 LION-B 440 490 15 15 3481
 98 GUAM-A 440 490 15 15 3481
 98 GUAM-B 440 490 15 15 3481
 98 PIKE-A 440 490 15 15 3481
 99 POGO-A 1225 1305 20 15 3380
 99 POGO-B 1225 1305 20 15 3380
 99 POGO-C 1225 1305 20 15 3380
 99 HULA-A 1225 1305 20 15 3380
 99 HULA-B 1225 1305 20 15 3380
 99 COOK-A 1225 1305 20 15 3380
 99 COOK-B 1225 1305 20 15 3380
 99 BOSS-A 1225 1305 20 15 3380
 99 BOSS-B 1225 1305 20 15 3380
 99 LION-A 1225 1305 20 15 3380
 99 LION-B 1225 1305 20 15 3380
 99 GUAM-A 1225 1305 20 15 3380
 99 GUAM-B 1225 1305 20 15 3380
 99 PIKE-A 1225 1305 20 15 3380

100 POGO-A 820 870 15 15 6439
 100 POGO-B 820 870 15 15 6439
 100 POGO-C 820 870 15 15 6439
 100 HULA-A 820 870 15 15 6439
 100 HULA-B 820 870 15 15 6439
 100 COOK-A 820 870 15 15 6439
 100 COOK-B 820 870 15 15 6439
 100 BOSS-A 820 870 15 15 6439
 100 BOSS-B 820 870 15 15 6439
 100 LION-A 820 870 15 15 6439
 100 LION-B 820 870 15 15 6439
 100 GUAM-A 820 870 15 15 6439
 100 GUAM-B 820 870 15 15 6439
 100 PIKE-A 820 870 15 15 6439
 101 POGO-A 860 925 5 15 5592
 101 POGO-B 860 925 5 15 5592
 101 POGO-C 860 925 5 15 5592
 101 HULA-A 860 925 5 15 5592
 101 HULA-B 860 925 5 15 5592
 101 COOK-A 860 925 5 15 5592
 101 COOK-B 860 925 5 15 5592
 101 BOSS-A 860 925 5 15 5592
 101 BOSS-B 860 925 5 15 5592
 101 LION-A 860 925 5 15 5592
 101 LION-B 860 925 5 15 5592
 101 GUAM-A 860 925 5 15 5592
 101 GUAM-B 860 925 5 15 5592
 101 PIKE-A 860 925 5 15 5592
 102 COOK-A 785 835 10 15 7586
 102 COOK-B 785 835 10 15 7586
 102 BOSS-A 785 835 10 15 7586
 102 BOSS-B 785 835 10 15 7586
 102 LION-A 785 835 10 15 7586
 102 LION-B 785 835 10 15 7586
 102 PIKE-A 785 835 10 15 7586
 103 COOK-A 1025 1105 20 15 4958
 103 COOK-B 1025 1105 20 15 4958
 103 BOSS-A 1025 1105 20 15 4958
 103 BOSS-B 1025 1105 20 15 4958
 103 LION-A 1025 1105 20 15 4958
 103 LION-B 1025 1105 20 15 4958
 103 PIKE-A 1025 1105 20 15 4958
 104 COOK-A 930 980 15 15 6227
 104 COOK-B 930 980 15 15 6227
 104 BOSS-A 930 980 15 15 6227
 104 BOSS-B 930 980 15 15 6227
 104 LION-A 930 980 15 15 6227
 104 LION-B 930 980 15 15 6227
 104 PIKE-A 930 980 15 15 6227
 105 COOK-A 620 670 10 15 6424
 105 COOK-B 620 670 10 15 6424
 105 BOSS-A 620 670 10 15 6424
 105 BOSS-B 620 670 10 15 6424

105 LION-A 620 670 10 15 6424
 105 LION-B 620 670 10 15 6424
 105 PIKE-A 620 670 10 15 6424
 106 COOK-A 730 810 20 15 4954
 106 COOK-B 730 810 20 15 4954
 106 BOSS-A 730 810 20 15 4954
 106 BOSS-B 730 810 20 15 4954
 106 LION-A 730 810 20 15 4954
 106 LION-B 730 810 20 15 4954
 106 PIKE-A 730 810 20 15 4954
 107 COOK-A 470 550 20 15 1231
 107 COOK-B 470 550 20 15 1231
 107 BOSS-A 470 550 20 15 1231
 107 BOSS-B 470 550 20 15 1231
 107 LION-A 470 550 20 15 1231
 107 LION-B 470 550 20 15 1231
 107 PIKE-A 470 550 20 15 1231
 108 COOK-A 1120 1170 15 15 1788
 108 COOK-B 1120 1170 15 15 1788
 108 BOSS-A 1120 1170 15 15 1788
 108 BOSS-B 1120 1170 15 15 1788
 108 LION-A 1120 1170 15 15 1788
 108 LION-B 1120 1170 15 15 1788
 108 PIKE-A 1120 1170 15 15 1788
 109 POGO-A 1260 1325 5 15 7558
 109 POGO-B 1260 1325 5 15 7558
 109 POGO-C 1260 1325 5 15 7558
 109 HULA-A 1260 1325 5 15 7558
 109 HULA-B 1260 1325 5 15 7558
 109 COOK-A 1260 1325 5 15 7558
 109 COOK-B 1260 1325 5 15 7558
 109 BOSS-A 1260 1325 5 15 7558
 109 BOSS-B 1260 1325 5 15 7558
 109 LION-A 1260 1325 5 15 7558
 109 LION-B 1260 1325 5 15 7558
 109 GUAM-A 1260 1325 5 15 7558
 109 GUAM-B 1260 1325 5 15 7558
 109 PIKE-A 1260 1325 5 15 7558
 110 POGO-A 1335 1385 15 15 3332
 110 POGO-B 1335 1385 15 15 3332
 110 POGO-C 1335 1385 15 15 3332
 110 HULA-A 1335 1385 15 15 3332
 110 HULA-B 1335 1385 15 15 3332
 110 COOK-A 1335 1385 15 15 3332
 110 COOK-B 1335 1385 15 15 3332
 110 BOSS-A 1335 1385 15 15 3332
 110 BOSS-B 1335 1385 15 15 3332
 110 LION-A 1335 1385 15 15 3332
 110 LION-B 1335 1385 15 15 3332
 110 GUAM-A 1335 1385 15 15 3332
 110 GUAM-B 1335 1385 15 15 3332
 110 PIKE-A 1335 1385 15 15 3332
 111 POGO-A 655 735 20 15 5670

111 POGO-B 655 735 20 15 5670
 111 POGO-C 655 735 20 15 5670
 111 HULA-A 655 735 20 15 5670
 111 HULA-B 655 735 20 15 5670
 111 COOK-A 655 735 20 15 5670
 111 COOK-B 655 735 20 15 5670
 111 BOSS-A 655 735 20 15 5670
 111 BOSS-B 655 735 20 15 5670
 111 LION-A 655 735 20 15 5670
 111 LION-B 655 735 20 15 5670
 111 GUAM-A 655 735 20 15 5670
 111 GUAM-B 655 735 20 15 5670
 111 PIKE-A 655 735 20 15 5670
 112 POGO-A 180 245 5 15 4114
 112 POGO-B 180 245 5 15 4114
 112 POGO-C 180 245 5 15 4114
 112 HULA-A 180 245 5 15 4114
 112 HULA-B 180 245 5 15 4114
 112 COOK-A 180 245 5 15 4114
 112 COOK-B 180 245 5 15 4114
 112 BOSS-A 180 245 5 15 4114
 112 BOSS-B 180 245 5 15 4114
 112 LION-A 180 245 5 15 4114
 112 LION-B 180 245 5 15 4114
 112 GUAM-A 180 245 5 15 4114
 112 GUAM-B 180 245 5 15 4114
 112 PIKE-A 180 245 5 15 4114
 113 POGO-A 25 90 5 15 6916
 113 POGO-B 25 90 5 15 6916
 113 POGO-C 25 90 5 15 6916
 113 HULA-A 25 90 5 15 6916
 113 HULA-B 25 90 5 15 6916
 113 COOK-A 25 90 5 15 6916
 113 COOK-B 25 90 5 15 6916
 113 BOSS-A 25 90 5 15 6916
 113 BOSS-B 25 90 5 15 6916
 113 LION-A 25 90 5 15 6916
 113 LION-B 25 90 5 15 6916
 113 GUAM-A 25 90 5 15 6916
 113 GUAM-B 25 90 5 15 6916
 113 PIKE-A 25 90 5 15 6916
 114 POGO-A 865 930 5 15 1895
 114 POGO-B 865 930 5 15 1895
 114 POGO-C 865 930 5 15 1895
 114 HULA-A 865 930 5 15 1895
 114 HULA-B 865 930 5 15 1895
 114 COOK-A 865 930 5 15 1895
 114 COOK-B 865 930 5 15 1895
 114 BOSS-A 865 930 5 15 1895
 114 BOSS-B 865 930 5 15 1895
 114 LION-A 865 930 5 15 1895
 114 LION-B 865 930 5 15 1895
 114 GUAM-A 865 930 5 15 1895

114 GUAM-B 865 930 5 15 1895
 114 PIKE-A 865 930 5 15 1895
 115 POGO-A 70 120 15 15 6864
 115 POGO-B 70 120 15 15 6864
 115 POGO-C 70 120 15 15 6864
 115 HULA-A 70 120 15 15 6864
 115 HULA-B 70 120 15 15 6864
 115 COOK-A 70 120 15 15 6864
 115 COOK-B 70 120 15 15 6864
 115 INDI-A 70 120 15 15 6864
 115 BOSS-A 70 120 15 15 6864
 115 BOSS-B 70 120 15 15 6864
 115 LION-A 70 120 15 15 6864
 115 LION-B 70 120 15 15 6864
 115 GUAM-A 70 120 15 15 6864
 115 GUAM-B 70 120 15 15 6864
 115 PIKE-A 70 120 15 15 6864
 116 POGO-A 895 975 20 15 1297
 116 POGO-B 895 975 20 15 1297
 116 POGO-C 895 975 20 15 1297
 116 HULA-A 895 975 20 15 1297
 116 HULA-B 895 975 20 15 1297
 116 COOK-A 895 975 20 15 1297
 116 COOK-B 895 975 20 15 1297
 116 INDI-A 895 975 20 15 1297
 116 BOSS-A 895 975 20 15 1297
 116 BOSS-B 895 975 20 15 1297
 116 LION-A 895 975 20 15 1297
 116 LION-B 895 975 20 15 1297
 116 GUAM-A 895 975 20 15 1297
 116 GUAM-B 895 975 20 15 1297
 116 PIKE-A 895 975 20 15 1297
 117 POGO-A 1175 1255 20 15 1177
 117 POGO-B 1175 1255 20 15 1177
 117 POGO-C 1175 1255 20 15 1177
 117 HULA-A 1175 1255 20 15 1177
 117 HULA-B 1175 1255 20 15 1177
 117 COOK-A 1175 1255 20 15 1177
 117 COOK-B 1175 1255 20 15 1177
 117 INDI-A 1175 1255 20 15 1177
 117 BOSS-A 1175 1255 20 15 1177
 117 BOSS-B 1175 1255 20 15 1177
 117 LION-A 1175 1255 20 15 1177
 117 LION-B 1175 1255 20 15 1177
 117 GUAM-A 1175 1255 20 15 1177
 117 GUAM-B 1175 1255 20 15 1177
 117 PIKE-A 1175 1255 20 15 1177
 118 POGO-A 560 640 20 15 7761
 118 POGO-B 560 640 20 15 7761
 118 POGO-C 560 640 20 15 7761
 118 HULA-A 560 640 20 15 7761
 118 HULA-B 560 640 20 15 7761
 118 COOK-A 560 640 20 15 7761

118 COOK-B 560 640 20 15 7761
 118 INDI-A 560 640 20 15 7761
 118 BOSS-A 560 640 20 15 7761
 118 BOSS-B 560 640 20 15 7761
 118 LION-A 560 640 20 15 7761
 118 LION-B 560 640 20 15 7761
 118 GUAM-A 560 640 20 15 7761
 118 GUAM-B 560 640 20 15 7761
 118 PIKE-A 560 640 20 15 7761
 119 POGO-A 485 535 10 15 5161
 119 POGO-B 485 535 10 15 5161
 119 POGO-C 485 535 10 15 5161
 119 HULA-A 485 535 10 15 5161
 119 HULA-B 485 535 10 15 5161
 119 COOK-A 485 535 10 15 5161
 119 COOK-B 485 535 10 15 5161
 119 INDI-A 485 535 10 15 5161
 119 BOSS-A 485 535 10 15 5161
 119 BOSS-B 485 535 10 15 5161
 119 LION-A 485 535 10 15 5161
 119 LION-B 485 535 10 15 5161
 119 GUAM-A 485 535 10 15 5161
 119 GUAM-B 485 535 10 15 5161
 119 PIKE-A 485 535 10 15 5161
 120 POGO-A 315 395 20 15 7646
 120 POGO-B 315 395 20 15 7646
 120 POGO-C 315 395 20 15 7646
 120 HULA-A 315 395 20 15 7646
 120 HULA-B 315 395 20 15 7646
 120 COOK-A 315 395 20 15 7646
 120 COOK-B 315 395 20 15 7646
 120 INDI-A 315 395 20 15 7646
 120 BOSS-A 315 395 20 15 7646
 120 BOSS-B 315 395 20 15 7646
 120 LION-A 315 395 20 15 7646
 120 LION-B 315 395 20 15 7646
 120 GUAM-A 315 395 20 15 7646
 120 GUAM-B 315 395 20 15 7646
 120 PIKE-A 315 395 20 15 7646
 121 INDI-A 625 675 10 15 6570
 121 GUAM-A 625 675 10 15 6570
 121 GUAM-B 625 675 10 15 6570
 121 REEF-A 625 675 10 15 6570
 122 INDI-A 605 655 10 15 3131
 122 GUAM-A 605 655 10 15 3131
 122 GUAM-B 605 655 10 15 3131
 122 REEF-A 605 655 10 15 3131
 123 INDI-A 160 240 20 15 7043
 123 GUAM-A 160 240 20 15 7043
 123 GUAM-B 160 240 20 15 7043
 123 REEF-A 160 240 20 15 7043
 124 INDI-A 205 255 15 15 7897
 124 GUAM-A 205 255 15 15 7897

124 GUAM-B 205 255 15 15 7897
 124 REEF-A 205 255 15 15 7897
 125 INDI-A 1125 1175 10 15 8670
 125 GUAM-A 1125 1175 10 15 8670
 125 GUAM-B 1125 1175 10 15 8670
 125 REEF-A 1125 1175 10 15 8670
 126 HULA-A 575 625 10 15 3669
 126 HULA-B 575 625 10 15 3669
 126 COOK-A 575 625 10 15 3669
 126 COOK-B 575 625 10 15 3669
 126 GUAM-A 575 625 10 15 3669
 126 GUAM-B 575 625 10 15 3669
 126 PIKE-A 575 625 10 15 3669
 127 HULA-A 210 260 15 15 6881
 127 HULA-B 210 260 15 15 6881
 127 COOK-A 210 260 15 15 6881
 127 COOK-B 210 260 15 15 6881
 127 GUAM-A 210 260 15 15 6881
 127 GUAM-B 210 260 15 15 6881
 127 PIKE-A 210 260 15 15 6881
 128 HULA-A 940 1005 5 15 7030
 128 HULA-B 940 1005 5 15 7030
 128 COOK-A 940 1005 5 15 7030
 128 COOK-B 940 1005 5 15 7030
 128 GUAM-A 940 1005 5 15 7030
 128 GUAM-B 940 1005 5 15 7030
 128 PIKE-A 940 1005 5 15 7030
 129 HULA-A 1085 1135 10 15 7332
 129 HULA-B 1085 1135 10 15 7332
 129 COOK-A 1085 1135 10 15 7332
 129 COOK-B 1085 1135 10 15 7332
 129 GUAM-A 1085 1135 10 15 7332
 129 GUAM-B 1085 1135 10 15 7332
 129 PIKE-A 1085 1135 10 15 7332
 130 HULA-A 190 240 10 15 1038
 130 HULA-B 190 240 10 15 1038
 130 COOK-A 190 240 10 15 1038
 130 COOK-B 190 240 10 15 1038
 130 GUAM-A 190 240 10 15 1038
 130 GUAM-B 190 240 10 15 1038
 130 PIKE-A 190 240 10 15 1038
 131 HULA-A 500 565 5 15 4141
 131 HULA-B 500 565 5 15 4141
 131 COOK-A 500 565 5 15 4141
 131 COOK-B 500 565 5 15 4141
 131 GUAM-A 500 565 5 15 4141
 131 GUAM-B 500 565 5 15 4141
 132 HULA-A 765 815 10 15 3728
 132 HULA-B 765 815 10 15 3728
 132 COOK-A 765 815 10 15 3728
 132 COOK-B 765 815 10 15 3728
 132 GUAM-A 765 815 10 15 3728
 132 GUAM-B 765 815 10 15 3728

133 HULA-A 1310 1360 10 15 5610
 133 HULA-B 1310 1360 10 15 5610
 133 COOK-A 1310 1360 10 15 5610
 133 COOK-B 1310 1360 10 15 5610
 133 GUAM-A 1310 1360 10 15 5610
 133 GUAM-B 1310 1360 10 15 5610
 134 HULA-A 1305 1370 5 15 4860
 134 HULA-B 1305 1370 5 15 4860
 134 COOK-A 1305 1370 5 15 4860
 134 COOK-B 1305 1370 5 15 4860
 134 GUAM-A 1305 1370 5 15 4860
 134 GUAM-B 1305 1370 5 15 4860
 135 HULA-A 905 955 10 15 5002
 135 HULA-B 905 955 10 15 5002
 135 COOK-A 905 955 10 15 5002
 135 COOK-B 905 955 10 15 5002
 135 GUAM-A 905 955 10 15 5002
 135 GUAM-B 905 955 10 15 5002
 136 POGO-A 815 880 5 15 1695
 136 POGO-B 815 880 5 15 1695
 136 POGO-C 815 880 5 15 1695
 136 COOK-A 815 880 5 15 1695
 136 COOK-B 815 880 5 15 1695
 136 INDI-A 815 880 5 15 1695
 136 LION-A 815 880 5 15 1695
 136 LION-B 815 880 5 15 1695
 136 GUAM-A 815 880 5 15 1695
 136 GUAM-B 815 880 5 15 1695
 136 PIKE-A 815 880 5 15 1695
 136 REEF-A 815 880 5 15 1695
 137 POGO-A 465 515 15 15 7532
 137 POGO-B 465 515 15 15 7532
 137 POGO-C 465 515 15 15 7532
 137 COOK-A 465 515 15 15 7532
 137 COOK-B 465 515 15 15 7532
 137 INDI-A 465 515 15 15 7532
 137 LION-A 465 515 15 15 7532
 137 LION-B 465 515 15 15 7532
 137 GUAM-A 465 515 15 15 7532
 137 GUAM-B 465 515 15 15 7532
 137 PIKE-A 465 515 15 15 7532
 137 REEF-A 465 515 15 15 7532
 138 POGO-A 535 585 15 15 7478
 138 POGO-B 535 585 15 15 7478
 138 POGO-C 535 585 15 15 7478
 138 COOK-A 535 585 15 15 7478
 138 COOK-B 535 585 15 15 7478
 138 INDI-A 535 585 15 15 7478
 138 LION-A 535 585 15 15 7478
 138 LION-B 535 585 15 15 7478
 138 GUAM-A 535 585 15 15 7478
 138 GUAM-B 535 585 15 15 7478
 138 PIKE-A 535 585 15 15 7478

138 REEF-A 535 585 15 15 7478
 139 POGO-A 250 300 15 15 1952
 139 POGO-B 250 300 15 15 1952
 139 POGO-C 250 300 15 15 1952
 139 COOK-A 250 300 15 15 1952
 139 COOK-B 250 300 15 15 1952
 139 INDI-A 250 300 15 15 1952
 139 LION-A 250 300 15 15 1952
 139 LION-B 250 300 15 15 1952
 139 GUAM-A 250 300 15 15 1952
 139 GUAM-B 250 300 15 15 1952
 139 PIKE-A 250 300 15 15 1952
 139 REEF-A 250 300 15 15 1952
 140 POGO-A 1200 1250 10 15 7964
 140 POGO-B 1200 1250 10 15 7964
 140 POGO-C 1200 1250 10 15 7964
 140 COOK-A 1200 1250 10 15 7964
 140 COOK-B 1200 1250 10 15 7964
 140 BOSS-A 1200 1250 10 15 7964
 140 BOSS-B 1200 1250 10 15 7964
 140 LION-A 1200 1250 10 15 7964
 140 LION-B 1200 1250 10 15 7964
 140 PIKE-A 1200 1250 10 15 7964
 141 POGO-A 565 615 10 15 1643
 141 POGO-B 565 615 10 15 1643
 141 POGO-C 565 615 10 15 1643
 141 COOK-A 565 615 10 15 1643
 141 COOK-B 565 615 10 15 1643
 141 BOSS-A 565 615 10 15 1643
 141 BOSS-B 565 615 10 15 1643
 141 LION-A 565 615 10 15 1643
 141 LION-B 565 615 10 15 1643
 141 PIKE-A 565 615 10 15 1643
 142 POGO-A 745 795 15 15 7443
 142 POGO-B 745 795 15 15 7443
 142 POGO-C 745 795 15 15 7443
 142 COOK-A 745 795 15 15 7443
 142 COOK-B 745 795 15 15 7443
 142 BOSS-A 745 795 15 15 7443
 142 BOSS-B 745 795 15 15 7443
 142 LION-A 745 795 15 15 7443
 142 LION-B 745 795 15 15 7443
 142 PIKE-A 745 795 15 15 7443
 143 POGO-A 1355 1420 5 15 5903
 143 POGO-B 1355 1420 5 15 5903
 143 POGO-C 1355 1420 5 15 5903
 143 COOK-A 1355 1420 5 15 5903
 143 COOK-B 1355 1420 5 15 5903
 143 BOSS-A 1355 1420 5 15 5903
 143 BOSS-B 1355 1420 5 15 5903
 143 LION-A 1355 1420 5 15 5903
 143 LION-B 1355 1420 5 15 5903
 143 PIKE-A 1355 1420 5 15 5903

144 POGO-A 1360 1410 15 15 5919
 144 POGO-B 1360 1410 15 15 5919
 144 POGO-C 1360 1410 15 15 5919
 144 COOK-A 1360 1410 15 15 5919
 144 COOK-B 1360 1410 15 15 5919
 144 INDI-A 1360 1410 15 15 5919
 144 BOSS-A 1360 1410 15 15 5919
 144 BOSS-B 1360 1410 15 15 5919
 144 LION-A 1360 1410 15 15 5919
 144 LION-B 1360 1410 15 15 5919
 144 PIKE-A 1360 1410 15 15 5919
 145 POGO-A 1170 1220 15 15 4261
 145 POGO-B 1170 1220 15 15 4261
 145 POGO-C 1170 1220 15 15 4261
 145 COOK-A 1170 1220 15 15 4261
 145 COOK-B 1170 1220 15 15 4261
 145 INDI-A 1170 1220 15 15 4261
 145 BOSS-A 1170 1220 15 15 4261
 145 BOSS-B 1170 1220 15 15 4261
 145 LION-A 1170 1220 15 15 4261
 145 LION-B 1170 1220 15 15 4261
 145 PIKE-A 1170 1220 15 15 4261
 146 POGO-A 590 640 10 15 5939
 146 POGO-B 590 640 10 15 5939
 146 POGO-C 590 640 10 15 5939
 146 COOK-A 590 640 10 15 5939
 146 COOK-B 590 640 10 15 5939
 146 INDI-A 590 640 10 15 5939
 146 BOSS-A 590 640 10 15 5939
 146 BOSS-B 590 640 10 15 5939
 146 LION-A 590 640 10 15 5939
 146 LION-B 590 640 10 15 5939
 146 PIKE-A 590 640 10 15 5939
 147 POGO-A 610 675 5 15 8650
 147 POGO-B 610 675 5 15 8650
 147 POGO-C 610 675 5 15 8650
 147 INDI-A 610 675 5 15 8650
 147 BOSS-A 610 675 5 15 8650
 147 BOSS-B 610 675 5 15 8650
 147 LION-A 610 675 5 15 8650
 147 LION-B 610 675 5 15 8650
 147 GUAM-A 610 675 5 15 8650
 147 GUAM-B 610 675 5 15 8650
 147 REEF-A 610 675 5 15 8650
 148 POGO-A 325 390 5 15 3661
 148 POGO-B 325 390 5 15 3661
 148 POGO-C 325 390 5 15 3661
 148 INDI-A 325 390 5 15 3661
 148 BOSS-A 325 390 5 15 3661
 148 BOSS-B 325 390 5 15 3661
 148 LION-A 325 390 5 15 3661
 148 LION-B 325 390 5 15 3661
 148 GUAM-A 325 390 5 15 3661

148 GUAM-B 325 390 5 15 3661
 148 REEF-A 325 390 5 15 3661
 149 POGO-A 1255 1305 10 15 4526
 149 POGO-B 1255 1305 10 15 4526
 149 POGO-C 1255 1305 10 15 4526
 149 INDI-A 1255 1305 10 15 4526
 149 BOSS-A 1255 1305 10 15 4526
 149 BOSS-B 1255 1305 10 15 4526
 149 LION-A 1255 1305 10 15 4526
 149 LION-B 1255 1305 10 15 4526
 149 GUAM-A 1255 1305 10 15 4526
 149 GUAM-B 1255 1305 10 15 4526
 149 PIKE-A 1255 1305 10 15 4526
 149 REEF-A 1255 1305 10 15 4526
 150 POGO-A 1055 1105 10 15 2206
 150 POGO-B 1055 1105 10 15 2206
 150 POGO-C 1055 1105 10 15 2206
 150 INDI-A 1055 1105 10 15 2206
 150 BOSS-A 1055 1105 10 15 2206
 150 BOSS-B 1055 1105 10 15 2206
 150 LION-A 1055 1105 10 15 2206
 150 LION-B 1055 1105 10 15 2206
 150 GUAM-A 1055 1105 10 15 2206
 150 GUAM-B 1055 1105 10 15 2206
 150 PIKE-A 1055 1105 10 15 2206
 150 REEF-A 1055 1105 10 15 2206
 151 POGO-A 740 790 10 15 5521
 151 POGO-B 740 790 10 15 5521
 151 POGO-C 740 790 10 15 5521
 151 HULA-A 740 790 10 15 5521
 151 HULA-B 740 790 10 15 5521
 151 COOK-A 740 790 10 15 5521
 151 COOK-B 740 790 10 15 5521
 151 INDI-A 740 790 10 15 5521
 151 LION-A 740 790 10 15 5521
 151 LION-B 740 790 10 15 5521
 151 PIKE-A 740 790 10 15 5521
 151 REEF-A 740 790 10 15 5521
 152 POGO-A 880 930 10 15 6966
 152 POGO-B 880 930 10 15 6966
 152 POGO-C 880 930 10 15 6966
 152 HULA-A 880 930 10 15 6966
 152 HULA-B 880 930 10 15 6966
 152 COOK-A 880 930 10 15 6966
 152 COOK-B 880 930 10 15 6966
 152 INDI-A 880 930 10 15 6966
 152 LION-A 880 930 10 15 6966
 152 LION-B 880 930 10 15 6966
 152 PIKE-A 880 930 10 15 6966
 152 REEF-A 880 930 10 15 6966
 153 POGO-A 120 170 10 15 8089
 153 POGO-B 120 170 10 15 8089
 153 POGO-C 120 170 10 15 8089

153 HULA-A 120 170 10 15 8089
 153 HULA-B 120 170 10 15 8089
 153 COOK-A 120 170 10 15 8089
 153 COOK-B 120 170 10 15 8089
 153 BOSS-A 120 170 10 15 8089
 153 BOSS-B 120 170 10 15 8089
 153 PIKE-A 120 170 10 15 8089
 154 POGO-A 300 350 10 15 6594
 154 POGO-B 300 350 10 15 6594
 154 POGO-C 300 350 10 15 6594
 154 HULA-A 300 350 10 15 6594
 154 HULA-B 300 350 10 15 6594
 154 COOK-A 300 350 10 15 6594
 154 COOK-B 300 350 10 15 6594
 154 BOSS-A 300 350 10 15 6594
 154 BOSS-B 300 350 10 15 6594
 154 PIKE-A 300 350 10 15 6594
 155 HULA-A 1270 1320 10 15 2805
 155 HULA-B 1270 1320 10 15 2805
 155 GUAM-A 1270 1320 10 15 2805
 155 GUAM-B 1270 1320 10 15 2805
 155 REEF-A 1270 1320 10 15 2805
 156 HULA-A 35 85 10 15 7928
 156 HULA-B 35 85 10 15 7928
 156 GUAM-A 35 85 10 15 7928
 156 GUAM-B 35 85 10 15 7928
 156 REEF-A 35 85 10 15 7928
 157 COOK-A 430 480 10 15 1267
 157 COOK-B 430 480 10 15 1267
 157 INDI-A 430 480 10 15 1267
 157 LION-A 430 480 10 15 1267
 157 LION-B 430 480 10 15 1267
 157 GUAM-A 430 480 10 15 1267
 157 GUAM-B 430 480 10 15 1267
 157 PIKE-A 430 480 10 15 1267
 157 REEF-A 430 480 10 15 1267
 158 COOK-A 830 895 5 15 8161
 158 COOK-B 830 895 5 15 8161
 158 INDI-A 830 895 5 15 8161
 158 LION-A 830 895 5 15 8161
 158 LION-B 830 895 5 15 8161
 158 GUAM-A 830 895 5 15 8161
 158 GUAM-B 830 895 5 15 8161
 158 PIKE-A 830 895 5 15 8161
 158 REEF-A 830 895 5 15 8161
 159 POGO-A 1130 1180 10 15 7461
 159 POGO-B 1130 1180 10 15 7461
 159 POGO-C 1130 1180 10 15 7461
 159 LION-A 1130 1180 10 15 7461
 159 LION-B 1130 1180 10 15 7461
 159 GUAM-A 1130 1180 10 15 7461
 159 GUAM-B 1130 1180 10 15 7461
 159 PIKE-A 1130 1180 10 15 7461

160 POGO-A 945 995 10 15 4899
 160 POGO-B 945 995 10 15 4899
 160 POGO-C 945 995 10 15 4899
 160 INDI-A 945 995 10 15 4899
 160 BOSS-A 945 995 10 15 4899
 160 BOSS-B 945 995 10 15 4899
 160 REEF-A 945 995 10 15 4899
 161 POGO-A 660 725 5 15 7965
 161 POGO-B 660 725 5 15 7965
 161 POGO-C 660 725 5 15 7965
 161 HULA-A 660 725 5 15 7965
 161 HULA-B 660 725 5 15 7965
 161 COOK-A 660 725 5 15 7965
 161 COOK-B 660 725 5 15 7965
 161 LION-A 660 725 5 15 7965
 161 LION-B 660 725 5 15 7965
 161 GUAM-A 660 725 5 15 7965
 161 GUAM-B 660 725 5 15 7965
 161 PIKE-A 660 725 5 15 7965
 162 POGO-A 1155 1220 5 15 2549
 162 POGO-B 1155 1220 5 15 2549
 162 POGO-C 1155 1220 5 15 2549
 162 HULA-A 1155 1220 5 15 2549
 162 HULA-B 1155 1220 5 15 2549
 162 COOK-A 1155 1220 5 15 2549
 162 COOK-B 1155 1220 5 15 2549
 162 INDI-A 1155 1220 5 15 2549
 162 BOSS-A 1155 1220 5 15 2549
 162 BOSS-B 1155 1220 5 15 2549
 162 GUAM-A 1155 1220 5 15 2549
 162 GUAM-B 1155 1220 5 15 2549
 162 PIKE-A 1155 1220 5 15 2549
 163 POGO-A 1100 1150 10 15 1805
 163 POGO-B 1100 1150 10 15 1805
 163 POGO-C 1100 1150 10 15 1805
 163 HULA-A 1100 1150 10 15 1805
 163 HULA-B 1100 1150 10 15 1805
 163 COOK-A 1100 1150 10 15 1805
 163 COOK-B 1100 1150 10 15 1805
 163 GUAM-A 1100 1150 10 15 1805
 163 GUAM-B 1100 1150 10 15 1805
 164 POGO-A 1070 1120 10 15 4145
 164 POGO-B 1070 1120 10 15 4145
 164 POGO-C 1070 1120 10 15 4145
 164 HULA-A 1070 1120 10 15 4145
 164 HULA-B 1070 1120 10 15 4145
 164 COOK-A 1070 1120 10 15 4145
 164 COOK-B 1070 1120 10 15 4145
 164 BOSS-A 1070 1120 10 15 4145
 164 BOSS-B 1070 1120 10 15 4145
 164 GUAM-A 1070 1120 10 15 4145
 164 GUAM-B 1070 1120 10 15 4145
 164 PIKE-A 1070 1120 10 15 4145

165 BOSS-A 1135 1200 5 15 7520
165 BOSS-B 1135 1200 5 15 7520
166 COOK-A 390 455 5 15 5126
166 COOK-B 390 455 5 15 5126
166 INDI-A 390 455 5 15 5126
166 BOSS-A 390 455 5 15 5126
166 BOSS-B 390 455 5 15 5126
166 GUAM-A 390 455 5 15 5126
166 GUAM-B 390 455 5 15 5126
166 REEF-A 390 455 5 15 5126

167 COOK-A 1380 1430 10 15 1816
167 COOK-B 1380 1430 10 15 1816
167 BOSS-A 1380 1430 10 15 1816
167 BOSS-B 1380 1430 10 15 1816
167 LION-A 1380 1430 10 15 1816
167 LION-B 1380 1430 10 15 1816
167 GUAM-A 1380 1430 10 15 1816
167 GUAM-B 1380 1430 10 15 1816
167 PIKE-A 1380 1430 10 15 1816
167 REEF-A 1380 1430 10 15 1816

APPENDIX E

The Results from Scheduling

7 POGO-A	89	103	14	20	7432011
20 POGO-A	266	282	16	20	4950011
27 POGO-A	331	347	16	20	8754011
32 POGO-A	397	412	15	20	3725011
38 POGO-A	460	474	14	20	2834011
46 POGO-A	576	591	15	20	7191011
57 POGO-A	656	672	16	20	8617011
65 POGO-A	718	733	15	20	1418011
72 POGO-A	777	791	14	20	5948011
81 POGO-A	880	896	16	20	6939011
87 POGO-A	944	958	14	20	7180011
96 POGO-A	980	996	16	20	8793011
111 POGO-A	1133	1148	15	20	8968011
121 POGO-A	1224	1240	16	20	3162011
127 POGO-A	1285	1301	16	20	8677011
4 POGO-A	25	30	5	15	6916000
20 POGO-A	180	185	5	15	4114000
28 POGO-A	235	245	10	15	6618000
36 POGO-A	300	310	10	15	6594000
39 POGO-A	362	367	5	15	3661000
62 POGO-A	535	550	15	15	7478000
67 POGO-A	606	616	10	15	5939000
102 POGO-A	911	921	10	15	6966000
126 POGO-A	1100	1110	10	15	1805000
131 POGO-A	1163	1173	10	15	7461000
135 POGO-A	1189	1194	5	15	2549000
149 POGO-A	1316	1321	5	15	7558000
164 POGO-A	1360	1375	15	15	5919000
838	5.8194444444E-01				
5 POGO-B	81	96	15	20	3405011
15 POGO-B	207	222	15	20	6194011
21 POGO-B	270	286	16	20	6141011
37 POGO-B	459	475	16	20	3577011
44 POGO-B	562	578	16	20	2000011
58 POGO-B	659	674	15	20	2485011
70 POGO-B	770	784	14	20	4024011
75 POGO-B	808	824	16	20	6871011
83 POGO-B	913	928	15	20	7355011
91 POGO-B	960	975	15	20	3593011
114 POGO-B	1137	1153	16	20	1229011
123 POGO-B	1236	1251	15	20	2028011
129 POGO-B	1298	1313	15	20	6182011
140 POGO-B	1384	1398	14	20	2686011
3 POGO-B	20	40	20	15	4117000
15 POGO-B	120	130	10	15	8089000
119 POGO-B	1030	1080	50	15	3274000
142 POGO-B	1200	1210	10	15	7964000
148 POGO-B	1266	1276	10	15	4526000
159 POGO-B	1335	1350	15	15	3332000
698	4.8472222222E-01				
17 POGO-C	225	240	15	20	5312011
19 POGO-C	262	277	15	20	2332011
34 POGO-C	437	453	16	20	1957011
40 POGO-C	479	495	16	20	2488011
43 POGO-C	559	575	16	20	7762011
55 POGO-C	643	658	15	20	7103011
69 POGO-C	761	776	15	20	4042011
79 POGO-C	847	863	16	20	3565011
84 POGO-C	929	945	16	20	1553011
92 POGO-C	967	983	16	20	6728011
100 POGO-C	1012	1028	16	20	5356011
120 POGO-C	1218	1234	16	20	8869011
126 POGO-C	1267	1282	15	20	3195011
132 POGO-C	1309	1325	16	20	3104011
137 POGO-C	1366	1382	16	20	2298011
7 POGO-C	45	75	30	15	8824000
13 POGO-C	100	140	40	15	7780000
57 POGO-C	510	520	10	15	5161000
71 POGO-C	610	615	5	15	8650000
77 POGO-C	673	693	20	15	5670000
94 POGO-C	815	820	5	15	1695000
99 POGO-C	878	883	5	15	5592000
100 POGO-C	898	903	5	15	1895000
137 POGO-C	1170	1185	15	15	4261000
805	5.5902777778E-01				
3 HULA-A	49	63	14	20	7714011
6 HULA-A	87	100	13	20	4921011
18 HULA-A	235	247	12	20	7177011
41 HULA-A	537	554	17	20	1324011
60 HULA-A	679	691	12	20	7241011
68 HULA-A	751	763	12	20	7318011
82 HULA-A	903	919	16	20	1211011
97 HULA-A	982	996	14	20	3915011
102 HULA-A	1031	1045	14	20	2755011

124 HULA-A 1238 1254 16 20 6762011
 1 HULA-A 10 15 5 15 8308000
 12 HULA-A 120 140 20 15 2110000
 22 HULA-A 190 200 10 15 1038000
 34 HULA-A 285 330 45 15 8928000
 46 HULA-A 370 405 35 15 6729000
 54 HULA-A 450 465 15 15 4781000
 58 HULA-A 495 510 15 15 3599000
 64 HULA-A 569 589 20 15 7761000
 66 HULA-A 604 614 10 15 3669000
 134 HULA-A 1150 1155 5 15 2541000
 156 HULA-A 1310 1320 10 15 5610000
 165 HULA-A 1365 1390 25 15 1225000
 735 5.1041666667E-01
 14 HULA-B 203 220 17 20 2193011
 52 HULA-B 620 637 17 20 7109011
 61 HULA-B 685 702 17 20 7989011
 74 HULA-B 787 804 17 20 8991011
 95 HULA-B 975 992 17 20 7183011
 101 HULA-B 1018 1031 13 20 8851011
 6 HULA-B 35 45 10 15 7928000
 25 HULA-B 235 250 15 15 6881000
 29 HULA-B 265 280 15 15 1608000
 75 HULA-B 652 662 10 15 2848000
 78 HULA-B 717 722 5 15 7965000
 85 HULA-B 740 750 10 15 5521000
 93 HULA-B 819 829 10 15 2569000
 95 HULA-B 844 859 15 15 6439000
 108 HULA-B 940 945 5 15 7030000
 125 HULA-B 1085 1095 10 15 7332000
 151 HULA-B 1275 1440 165 15 1702000
 653 4.5347222222E-01
 13 COOK-A 170 181 11 20 3288011
 29 COOK-A 344 358 14 20 1576011
 36 COOK-A 456 469 13 20 3535011
 67 COOK-A 736 752 16 20 5160011
 73 COOK-A 780 790 10 20 4383011
 86 COOK-A 943 956 13 20 4139011
 134 COOK-A 1328 1344 16 20 4581011
 38 COOK-A 375 395 20 15 7646000
 45 COOK-A 416 436 20 15 5283000
 59 COOK-A 500 505 5 15 4141000
 91 COOK-A 805 815 10 15 7586000
 104 COOK-A 905 915 10 15 5002000
 115 COOK-A 990 1030 40 15 6938000
 118 COOK-A 1085 1105 20 15 4958000
 129 COOK-A 1120 1135 15 15 1788000
 144 COOK-A 1225 1245 20 15 3380000
 152 COOK-A 1280 1295 15 15 4638000
 155 COOK-A 1359 1364 5 15 4860000
 578 4.0138888889E-01
 23 COOK-B 286 301 15 20 4599011
 63 COOK-B 697 710 13 20 5409011

5 COOK-B 30 55 25 15 8369000
 44 COOK-B 360 660 300 15 8438000
 153 COOK-B 1295 1310 15 15 7069000
 453 3.1458333333E-01
 12 INDI-A 121 136 15 20 7388011
 28 INDI-A 334 347 13 20 1227011
 31 INDI-A 385 396 11 20 7571011
 48 INDI-A 581 592 11 20 5051011
 113 INDI-A 1136 1152 16 20 5114011
 9 INDI-A 55 85 30 15 2118000
 49 INDI-A 411 426 15 15 5656000
 52 INDI-A 441 451 10 15 1267000
 63 INDI-A 540 555 15 15 8157000
 68 INDI-A 620 645 25 15 7105000
 69 INDI-A 660 680 20 15 1958000
 87 INDI-A 755 770 15 15 2116000
 105 INDI-A 925 945 20 15 5187000
 109 INDI-A 960 970 10 15 4899000
 112 INDI-A 985 1005 20 15 7492000
 113 INDI-A 1020 1040 20 15 3061000
 136 INDI-A 1167 1187 20 15 7107000
 143 INDI-A 1220 1235 15 15 3149000
 157 INDI-A 1320 1330 10 15 8971000
 161 INDI-A 1345 1380 35 15 8988000
 671 4.6597222222E-01
 10 BOSS-A 101 151 50 20 7477011
 16 BOSS-A 210 219 9 20 6111011
 33 BOSS-A 422 439 17 20 3181011
 42 BOSS-A 548 588 40 20 6369011
 62 BOSS-A 689 704 15 20 7258011
 66 BOSS-A 730 745 15 20 2524011
 105 BOSS-A 1076 1086 10 20 1673011
 112 BOSS-A 1135 1146 11 20 3270011
 128 BOSS-A 1296 1341 45 20 8912011
 136 BOSS-A 1365 1383 18 20 7604011
 2 BOSS-A 15 30 15 15 3800000
 19 BOSS-A 170 180 10 15 7986000
 32 BOSS-A 265 280 15 15 6755000
 37 BOSS-A 305 330 25 15 8183000
 51 BOSS-A 454 474 20 15 2190000
 56 BOSS-A 508 528 20 15 1231000
 76 BOSS-A 650 665 15 15 6355000
 92 BOSS-A 795 810 15 15 4978000
 98 BOSS-A 845 865 20 15 4947000
 110 BOSS-A 950 965 15 15 6617000
 111 BOSS-A 980 990 10 15 4822000
 122 BOSS-A 1101 1111 10 15 4145000
 132 BOSS-A 1161 1166 5 15 7520000
 145 BOSS-A 1235 1240 5 15 7612000
 146 BOSS-A 1255 1275 20 15 3242000
 875 6.0763888889E-01
 26 BOSS-B 316 321 5 20 7578011
 45 BOSS-B 565 577 12 20 7808011

50 BOSS-B 606 617 11 20 6791011
 59 BOSS-B 663 675 12 20 3086011
 117 BOSS-B 1166 1176 10 20 5276011
 130 BOSS-B 1304 1315 11 20 3355011
 144 BOSS-B 1416 1433 17 20 8333011
 14 BOSS-B 115 150 35 15 3833000
 23 BOSS-B 200 240 40 15 6364000
 42 BOSS-B 345 360 15 15 8649000
 53 BOSS-B 440 455 15 15 3481000
 60 BOSS-B 515 520 5 15 7979000
 72 BOSS-B 632 642 10 15 6424000
 81 BOSS-B 690 715 25 15 8977000
 83 BOSS-B 730 750 20 15 4954000
 96 BOSS-B 825 840 15 15 2463000
 114 BOSS-B 980 995 15 15 6230000
 117 BOSS-B 1010 1020 10 15 7213000
 123 BOSS-B 1075 1120 45 15 5522000
 160 BOSS-B 1340 1385 45 15 8983000
 708 4.9166666667E-01
 22 LION-A 280 292 12 20 8747011
 64 LION-A 704 718 14 20 2322011
 71 LION-A 771 785 14 20 7580011
 93 LION-A 968 981 13 20 7945011
 104 LION-A 1060 1073 13 20 2855011
 110 LION-A 1118 1132 14 20 5919011
 116 LION-A 1160 1174 14 20 6669011
 122 LION-A 1233 1247 14 20 5804011
 135 LION-A 1347 1359 12 20 2208011
 143 LION-A 1403 1419 16 20 2766011
 16 LION-A 140 175 35 15 6928000
 26 LION-A 225 255 30 15 1774000
 41 LION-A 340 640 300 15 4848000
 84 LION-A 735 780 45 15 8031000
 90 LION-A 800 810 10 15 8414000
 106 LION-A 930 945 15 15 6227000
 121 LION-A 1088 1098 10 15 2206000
 139 LION-A 1189 1209 20 15 8492000
 154 LION-A 1300 1315 15 15 8858000
 163 LION-A 1374 1379 5 15 5903000
 971 6.7430555556E-01
 11 LION-B 103 118 15 20 2526011
 30 LION-B 346 358 12 20 7725011
 51 LION-B 616 629 13 20 2398011
 85 LION-B 942 956 14 20 7482011
 98 LION-B 996 1009 13 20 6158011
 108 LION-B 1091 1105 14 20 8807011
 119 LION-B 1203 1216 13 20 7417011
 125 LION-B 1252 1266 14 20 3982011
 142 LION-B 1398 1414 16 20 4096011
 17 LION-B 155 200 45 15 6102000
 47 LION-B 375 385 10 15 7331000
 61 LION-B 520 540 20 15 2618000
 65 LION-B 565 575 10 15 1643000

82 LION-B 690 740 50 15 4849000
 101 LION-B 875 885 10 15 8569000
 116 LION-B 1024 1029 5 15 6659000
 120 LION-B 1044 1059 15 15 4169000
 158 LION-B 1330 1375 45 15 3317000
 649 4.5069444444E-01
 2 GUAM-A 40 50 10 20 3807011
 39 GUAM-A 464 481 17 20 1106011
 54 GUAM-A 631 643 12 20 6379011
 78 GUAM-A 846 863 17 20 3372011
 99 GUAM-A 999 1009 10 20 2075011
 107 GUAM-A 1082 1094 12 20 3892011
 118 GUAM-A 1176 1189 13 20 6444011
 145 GUAM-A 1420 1432 12 20 7824011
 24 GUAM-A 205 220 15 15 7897000
 27 GUAM-A 235 245 10 15 8778000
 30 GUAM-A 260 275 15 15 1952000
 50 GUAM-A 410 430 20 15 7831000
 73 GUAM-A 658 668 10 15 6570000
 88 GUAM-A 760 805 45 15 2937000
 97 GUAM-A 878 883 5 15 8161000
 124 GUAM-A 1109 1134 25 15 1614000
 140 GUAM-A 1204 1219 15 15 5033000
 138 GUAM-A 1234 1254 20 15 1177000
 162 GUAM-A 1350 1370 20 15 4284000
 628 4.3611111111E-01
 1 GUAM-B 33 44 11 20 1208011
 24 GUAM-B 288 302 14 20 8990011
 109 GUAM-B 1115 1127 12 20 1888011
 10 GUAM-B 70 85 15 15 6864000
 18 GUAM-B 160 180 20 15 7043000
 70 GUAM-B 605 615 10 15 3131000
 74 GUAM-B 635 655 20 15 7686000
 89 GUAM-B 765 775 10 15 3728000
 107 GUAM-B 935 985 50 15 5724000
 130 GUAM-B 1142 1152 10 15 8670000
 133 GUAM-B 1167 1182 15 15 3192000
 367 2.5486111111E-01
 25 PIKE-A 314 330 16 20 7165011
 56 PIKE-A 651 667 16 20 8971011
 88 PIKE-A 948 961 13 20 5330011
 103 PIKE-A 1032 1048 16 20 6962011
 106 PIKE-A 1078 1092 14 20 5352011
 131 PIKE-A 1307 1323 16 20 1637011
 8 PIKE-A 50 55 5 15 4483000
 33 PIKE-A 270 285 15 15 6115000
 35 PIKE-A 345 350 5 15 5218000
 43 PIKE-A 365 390 25 15 6853000
 79 PIKE-A 682 692 10 15 4345000
 86 PIKE-A 745 760 15 15 7443000
 103 PIKE-A 895 915 20 15 1297000
 127 PIKE-A 1107 1127 20 15 5226000
 128 PIKE-A 1142 1157 15 15 5886000

141 PIKE-A 1195 1215 20 15 4374000
147 PIKE-A 1245 1280 35 15 6809000
561 3.89583333333E-01
4 REEF-A 51 67 16 20 3942011
9 REEF-A 98 114 16 20 1275011
35 REEF-A 445 460 15 20 3450011
77 REEF-A 842 857 15 20 6070011
80 REEF-A 877 892 15 20 1806011
115 REEF-A 1156 1171 15 20 8375011
133 REEF-A 1321 1336 15 20 7235011

139 REEF-A 1381 1397 16 20 2001011
21 REEF-A 185 220 35 15 1002000
31 REEF-A 255 295 40 15 2381000
40 REEF-A 335 365 30 15 8579000
48 REEF-A 390 395 5 15 5126000
55 REEF-A 475 490 15 15 7532000
80 REEF-A 670 710 40 15 1317000
150 REEF-A 1270 1280 10 15 2805000
563 3.90972222222E-01

BIBLIOGRAPHY

1. Pinedo, Michael. *Scheduling: Theory, Algorithms, and System*. New Jersey: A Simon & Schuster, 1995.
2. Gooley, Capt Tim. *Automating the Satellite Range Scheduling Process*, MS Thesis, AFIT/GOR/ENS/93M-06. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1993.
3. Schalck, Capt Stanley Michael. *Automating Satellite Range Scheduling*, MS Thesis, AFIT/GSO/ENS/93D-14. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1993.
4. Parish, Capt Donald Arthur. *A Genetic Algorithm Approach to Automating Satellite Range Scheduling*, MS Thesis, AFIT/GOR/ENS/94M-10. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1993.
5. Whitely, Darrel. "GENITOR: a different generic algorithm." *Proceeding of the Rocky Mountain Conference on Artificial Intelligence*. 118-130. 1988.
6. Holland, John. *Adaptation in Natural and artificial System*. Ann Arbor: University of Michigan Press, 1975. Reprinted by MIT Press, Cambridge, MA, 1992.
7. Michalewicz, Zbigniew. *Genetic Algorithms + Data structure = Evolution Programs*. New York: Springer-Verlag, 1992.

Vita

Captain Kwangho Jang [REDACTED] He graduated from Soungnam High School, Seoul in 1986 and attended the Korean Military Academy, graduating with a Bachelor of Science Degree in Electrical Engineering in March 1990. Upon graduation, he successfully completed Officers Basic Course at Kwangjoo to become a platoon leader. His first assignment was at the Third Infantry Division, as a platoon leader. After that he served as an aid de camp of the division commander. He entered the Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, in May 1994. He married Kyoungmin Kim in 1993. The couple has a son, Jun.

[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 1996		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE The Capacity of The Air Force Satellite Control Network			5. FUNDING NUMBERS	
6. AUTHOR(S) Kwangho Jang, Captain, ROKA				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-6583			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GSO/ENS/96D-01	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The daily mission objective of the Air Force Satellite Control Network (AFSCN) is to support communication with satellite systems. It is critical that the AFSCN operate 24 hours a day, seven days a week. Previous work on the satellite range scheduling problem has successfully scheduled over 90 percent of the satellite support requests. This research investigates the capacity of the AFSCN using an available satellite scheduling algorithm. This research has three objectives. The first objective is to be able to generate sample data sets which represent a day of satellite support requests for low, medium, and high altitude satellites. The second research objective is to schedule the satellite support requests in the sample data sets. The third objective is to determine an upper bound on the number of support requests which can be supported by the AFSCN. Based on the reported results, the AFSCN is able to support around 175 low altitude satellite support requests and 250 medium/high altitude satellite support requests. At this level of demand, the scheduling algorithm is able to schedule approximately 90 percent of the satellite support requests.				
14. SUBJECT TERMS Random sample generation, Scheduling, Mixed Integer Programming, Satellite Support			15. NUMBER OF PAGES 137	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	PR - Project
G - Grant	TA - Task
PE - Program Element	WU - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."

DOE - See authorities.

NASA - See Handbook NHB 2200.2.

NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.

DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

NASA - Leave blank.

NTIS - Leave blank.

Block 13. Abstract. Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (*NTIS only*).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.