

User's Manual for Computer Software

Table of Contents
for
User's Manual for Computer Software

I. User's Manual for Computer Software	1
A. Software installation	1
B. Overview of software	2
C. The Lowry land-use model	4
D. Locating-routing for scheduled transportation	7
E. Remote sensing and geographic information system	10
F. Multiple travelling-salesmen-problem	18
G. A location-routing delivery-model	20
H. Derivation-allocation model for state parks	22
I. Garin-Lowry model	23
II. References	24

I. User's Manual for Computer Software

A. Software installation

Important!:

You must have at least 3 Megabytes of hard drive space available to install this disk. If you do not have 3 Megabytes available you must free up space or the installation will not work!

To install the contents of this disk on your hard drive, you must know which drive you are going to install on. For most users this will be the C: drive. First go to the disk drive where the floppy disk included with the book is located. For example, if the floppy is in the a: drive, type A:. Then type INSTALL followed by the drive you wish to install onto. For example, if you want to install on the C: drive, type INSTALL C:. This will create a directory on the C: drive named book where the files will extract to. The file readme.txt in C:\BOOK will explain the contents of the directory. The install program will leave you in the directory C:\BOOK when it completes. To return to this directory later, type CD C:\BOOK. To view the contents of the readme file type TYPE README.TXT | MORE.

For more experienced users, installation can be accomplished manually by simply copying the BOOK.EXE file to a directory on the hard drive and typing book to run the self-extracting zipped file.

The install program and all of the programs on this disk are designed to be run in DOS. In most cases the programs on this disk can be run from a Windows machine in a DOS window, but if a program won't run in a window the problem can be solved by booting the machine directly into DOS mode.

B. Overview of software

This disk is to provide the readers with sample software to supplement the discussions in the book. The platform is an IBM-compatible Personal Computer (PC), which at the time of this writing, is the most widely available computing machinery. By adopting this platform, it is expected that the accompanying disk will find its use among the largest number of readers. To view the ASCII text files in these disks, we suggest using a 10-point font format.

The disk is dedicated to specialized software for processing and analyzing spatial-temporal information. All of the software packages were developed by the author and his colleagues. The origins of these codes are duly acknowledged in the documentation, using a bibliographic referencing system adopted in the book. For example, a program to analyze remote-sensing images is stored under the SPACE directory. It was developed by Dr T S Kelso at the Air Force Institute of Technology when he worked with the author on related teaching and research. We have officially obtained his permission to use the program here. Software implementations of some of the facility-location and land-use models discussed in this book are contained in these additional directories, each of which has its own README files containing acknowledgments and operational instructions:

LOWRY - the traditional Lowry land-use model
RISE - a heuristic location-routing model for scheduled transportation
SPACEFIL - a heuristic multiple travelling-salesmen program
SPANFRST - a heuristic location-routing program for small-package deliveries
STATEPRK - an activity derivation-allocation model
YICHAN - a disaggregate/bifurcation implementation of the Garin-Lowry model

While the book introduces the various analysis techniques in a pedagogic fashion, we strive to provide the practical implementations here in this disk. The programs are therefore no longer purely for the classroom, they have real potentials among everyday, operational use.

These philosophies are followed in the preparation of this disk:

1. In order to provide the widest dissemination possible, all files in this disk are ASCII-text files (as mentioned). Where possible, both source codes and executable codes are given--mainly for the ease of execution and modification by the users.
2. We strive to provide stand-alone programs that do not require supporting software, including language compilers. While references are made to supporting software for extended use of some of the programs, all programs are self-contained and they have been developed or refined by the author and his associates.
3. Sample datasets are provided to allow demonstration of the software. While 'toy' problems are often used for introduction, most of these data are drawn from real-world case studies which are discussed in the main body of this book.
4. It is assumed that the user of this disk has acquired basic knowledge of the DOS operating system on a PC. S/he knows how to navigate between files and can discern the difference between a dataset file, an executable code and a source code. S/he also realizes that an executable code can be run by typing in the name of the file (without the extension) at a DOS prompt--with the necessary data. Accompanying this README file are two files providing information on how to compile a source code for execution.

The programs are distributed after extensively testing by the author and his associates. But with any computer programs, there is no guarantee that they are fully debugged. It is impossible for the author, with his limited

resources, to provide any programming support for these programs. But the author is keenly interested in and would appreciate any feedback from the users regarding their experiences in using these programs. The objective is to allow distribution of an improved software package at a future date. To provide your comments, simply contact:

ychan@alum.mit.edu

Now to view the program files again, please refer to the BOOK subdirectory. Have fun!

Acknowledgement

The author likes to acknowledge the assistance of Brad A. Smith. Mr. Smith was responsible for the compilation of the various software into a consistent format, organizing and test running all the programs on this disk.

C. The Lowry land-use model

As described in the "Activity-allocation" chapter of the book, the **LOWRY MODEL** generates and allocates population and employment among subdivisions in a study area. This directory contains the computer program that implements the classic Lowry Model.

1. This directory contains the FORTRAN source code - "lmlowprg.for", the IBM-PC executable code - "lmlowprg.exe" and the three data files - "lowry0.dat", "lowry1.dat", and "lowry2.dat".
2. The original FORTRAN Lowry-source-code, obtained from the University of Pennsylvania and subsequently modified at Pennsylvania State University, Washington State University and the Air Force Institute of Technology, was originally written using punch-card and line-printer-output commands. Here the modified "lmlowprg.for" uses files for input/output. To prevent printing of line printer carriage-control characters, format statements were modified. For example - "1H1"- new line was changed to "/"- skip line. "1H0"- double-line-spacing was changed to "1H "- blank.
3. Because the output was originally formatted for a 132-wide-character line printer, the output file--when viewed in an 80-character-wide mode--will show some line-wraparound. Therefore it is best to view the output file in an editor that has a lineprinter font. Also, printing the output file is best done in a lineprinter font.
4. To install and run the Lowry program on any PC:
 - a) copy "lmlowprg.exe", "lowry0.dat", "lowry1.dat", and "lowry2.dat" to your working directory.
 - b) To run the Lowry program, enter "lmlowprg". The program will prompt you to enter the input-data file and the output-data file.

Before we begin to show how to prepare the inputs to the program, here is a recap of the files included with the Lowry model:

LMLOWPRG.FOR -- FORTRAN source-listing of the model
LMLOWPRG.EXE -- Executable code of the program
YORK.DAT -- Base-year and forecast land-use/activity-distribution data for York, Pennsylvania, including model replication of base-year data. (NOTE: this file is for reference only and not meant to be processed by LMLOWPRG.EXE).
LOWRY0.DAT -- Base-year travel- and activity-distribution data for model-replication test
LOWRY1.DAT -- Forecast-year data, where the downtown-employment growth has been changed
LOWRY2.DAT -- Forecast-year data, where not only the downtown- employment has changed, but that a new uniform 25-miles/hour (40 km/h) Personal-Rapid-Transit Transportation-system has been implemented throughout the study area.

A FORTRAN PROGRAM OF THE LOWRY MODEL

The Lowry model listed here in this disk was tested extensively on a 42-zone York, Pennsylvania area for various experiments. To assist the reader in understanding the model, please refer to the discussion of the model in the "Activity allocation" chapter of this book. The following provides liberal annotations throughout the input stream for the programmer to run the program.

To clarify the operation of the program, we have highlighted the data-input format by the four groups of data required--a process judged to be more informative than extensive user-manuals. For a more complete documentation of the program, please refer to a workbook by Putman (1972), referenced in the book. The first set of inputs, as documented in all three data sets-- LOWRY0.DAT, LOWRY1.DAT and LOWRY2.DAT--are the "Run parameters", including the number of iterations and the coefficients required by the model. Specifically,

CONTROL PARAMETERS: The input of control-parameters, or the 'switches' that govern the mode in which the program is to be executed, is the same for both the base-year-replication input (LOWRY0.DAT) and the forecast-year input (LOWRY1.DAT, LOWRY2.DAT) with the exception of the parameter MAX which indicates the number of concentric rings used in the final time-ring matrix. For example, the number is often increased for a forecast year, say from 29 minutes maximum trip-time to 44 minutes, reflecting the slower travel speed of 25 miles/hour, and hence a longer trip-time (see LOWRY2.DAT when the travel speed is uniformly reduced to 25 miles per hour). Here are details of the line-by-line input for the control parameters:

Line 1:

L1 is a print command for the DPOP, DRET input data, where DPOP is the denominator of the population-potential-function, and DRET is the denominator of retail-potential-function (see the "Model calibration" section of the "Activity allocation" chapter for an explanation.) For the three datasets provided here--LOWRY0.DAT, LOWRY1.DAT, and LOWRY2.DAT--for example, 0 indicates that no printing is requested for these intermediate-output values. L2 is the number-of-iterations for the employment loop. For example, 4 is a typical number used in our datasets. L3 is the number of iterations for the population loop, shown in our datasets as 5. L4 is the number of grand-loops, shown as 4. L5 is the print command for intermediate output of activity summary tables; 0 means no and 1 means yes. L6 is the command to compute population-potential and begin the employment loop, where 1 means yes and 0 means no. M is the number-of-zones, shown here as 42 throughout all 3 datasets. N is the number-of-retail-trade-classes, shown here as 1 throughout. Other commands on line 1 are run IDs, output-type commands and the MAX parameter mentioned above.

Line 2:

K is the retail-trade-class under consideration. A(K) is the retail-employment-per-household, where K is the index of the retail-trade-class as mentioned immediately above. For our runs the parameter has been set as 0.470. C(K) is the residence-weight-factor, set as 0.900, which implies a complementary work-weight-factor of 0.1000. EC(K) is the retail-employment-per-thousand-square-feet, set as 1.01 here. ZS(K) is the minimum-efficient-size of retail-unit in number-of-employees, which is set as 10 here. The rest are potential-functions-parameters of the frequency-distribution of shopping-trips as defined in the "Model calibration" section of the "Activity allocation" chapter. These alpha, beta and gamma parameters have been set here as 55.3536, 14.6009, and 0.6651 respectively.

Line 3:

FC is the household-per-labor-force-member, set as 0.674. According to the function specified in the "Model calibration" section of the "Activity allocation" chapter, PE is the potential-function-power u, set as 0.1213; COEFF is the coefficient-of-work-trip-frequency-distribution, set as 1.664.

The *second dataset* contains activity and land-use data for such uses as residential and commercial, including these quantities, with land-use coded in thousand-of-square-feet (11 111 m²):

```
--total land,
--unusable land,
--basic land use,
--retail land,
--residential land,
--total employment,
--basic employment,
--retail employment,
```

--population potential (in number of households),
--number of household/1000 square feet
--maximum households/1000 square feet.

The *third dataset* is a square matrix documenting the travel time, in tenths of a minute, between two zones. Reference the 42-zone York map shown in the "Activity allocation" chapter, where we can interpret the 42x42 interzonal travel-time table. Notice the diagonal elements have zero entries as expected, since they represent the intrazonal travel-time which by definition is set as zero in our case. Not surprisingly, the matrix is symmetric with respect to the diagonal, suggesting the same amount of time to get from zone *i* to zone *j* as it is from *j* to *i*.

The last table represents the potential-ring-function set, the entries of which show the number of zones reachable with one minute of travel-time from the zone concerned. Notice this information is required for operationalizing the retail-distribution-function and the work-trip-distribution-function as shown in the "Model calibration" section of the chapter. For example, the entry in the 12th row and 14th column says that a radius of 14-minutes away, there are three additional zones reachable from zone-12 within an additional unit of travel-time. The base-year dataset LOWRY1.DAT has a potential-ring-function set measuring 42 by 29, since the maximum travel time is 29-minutes in the base year.

While the code documented here may not be the state-of-the-art in activity generation and allocation, it has its pedagogic importance in placing more recent models in proper perspective. For this reason, we have included example datasets in addition to the listings for those that want to have hands-on experience with such a classic model.

The intent of this program, once again, is simply to provide a "bread board" for experimentation. The datasets given are as important as (or more important than) the computer program, since more advanced computational procedures exist, but many first-time users are often at-a-loss as to how to assemble a quasi-realistic dataset for experimentation. Those interested in more advanced models should note that Foot (1978) documented a FORTRAN program of the Garin-Lowry model. Putman (1983) put forth an integrated model combining land-use and transportation. Yi and Chan (1988) extended the Garin-Lowry model to a disaggregate/bifurcation format that is included in this book and this computer disk as well, as discussed in the "Chaos, catastrophe, bifurcation" chapter. Now have fun with the numbers!

D. Locating-routing for scheduled transportation

RISE stands for Route Improvement Synthesis and Evaluation. It is a vehicle-routing and commodity-flow model developed by Y. Chan at MIT and subsequently extended at Pennsylvania State University at University Park, Washington State University, and the Air Force Institute of Technology at Wright-Patterson AFB, Ohio. The program is described in Chan (1974) and Chan (1979). The model is also explained and expanded in the "Location-routing" Chapter of the book.

This directory, RISE, contains:

- 1) the RISE Fortran source-program file - RISE.FOR
- 2) Rise executable file - RISE.EXE
- 3) three dataset files - RISE1.DAT, RISE2.DAT, RISE3.DAT
- 4) narrative on possible uses and description of the Rise program, File - USES.TXT
- 5) Echo print on American-Airlines-case-study data-input-stream, File- CASE.TXT. While it needs to be encoded in the input format, this can serve as an additional dataset if desired.
- 6) additional information on output-interpretation subroutines contained in the Rise Program, File - SUBROUT.TXT

To run the RISE program:

- 1) Copy RISE.EXE and the three RISE?.DAT files to your operating directory.
- 2) Enter "rise" at the DOS prompt and answer prompts for data file and output file.
- 3) View the output file in an editor that can handle 132-character-wide lines.

SOFTWARE DESCRIPTION

Inputs and Outputs

A package of programs has been developed for structuring scheduled-transportation route-networks. The following is a functional specification of the inputs, the variables and the outputs of the program package:

Given

- travel demand as a function of nonstop-, one-stop-, or two-stop- routings
- route specifications regarding intermediate stops and route geometry in general
- specification of the minimal level-of-service for a origin-destination pair in order to face up to route competition
- fleet specification (i.e., vehicle capacity, speed and range)
- the network under study (i.e., the specification of terminals and inter-terminal distances)
- revenues and cost functions.

Decision Variables

- route acceptance or rejection
- realized demand (i.e., that part of the potential demand actually served)
- route (or segment) traffic-flow identified by origin and destination
- assignment of vehicle type(s) to routes.

Solution

- route-network structure
- passenger or freight traffic-flow and routing
- realized demand
- route frequency
- fleet-type assignment

- fleet-size requirement.

Here are some detailed guidelines for the users in preparing the input data:

RISE1.DAT contains a test dataset intended to generate a star-shape route network

RISE2.DAT contains a test dataset intended to generate a two-loop route network

RISE3.DAT contains a test dataset intended to generate an open-loop route network

The data-input format can be summarized as follows, with examples drawn from RISE1.DAT:

Group 1 - VEHICLE-FLEET PARAMETERS

line 1:

Input-format specification, the example is (2I10,4F10.3) in FORTRAN.

line 2:

--Vehicle type 1, 2, 3 etc. (Only 1 type of vehicle allowed in the present version of the program)

--Capacity of vehicle (30 seats in this case),

--Speed coefficients in the form of (time in min.)=(ground-time coeff)+(speed-coeff in min/mi)(distance) and the example is (minutes)=50+0.3(dist in miles)

line 3:

blank-line delimiter

Group 2 - SYSTEM RUN PARAMETERS

line 1:

Input-format specification (4I10,2F10.5 in this example)

line 2:

--Vehicle type (1 in this case)

--No.-of-intermediate-stops in the longest route (2 in our case)

--Degree-of-optimism in the demand forecast (0-least, 3-most); in the present version of the program the demand is taken exactly as input, the code in this case is therefore immaterial.

--Type of origin-destination-pair market (0 means non-competitive, and 1 means competitive); again, this program does not have this feature implemented, the nonfunctional entry of 1 is input in this example.

--Ratio of effective-to-actual-capacity (0.83333333 entered here, suggesting only 83-percent of capacity is used for scheduling purposes, allowing for a "slack" of 17-percent, which accounts for all uncertainties such as overbooking.)

--Profit-margin above direct-operating-cost; set here as 0-percent, meaning that we will schedule a dispatch as long as it is break-even.

Group 3 - SPECIFICATION OF STOPS AND VEHICLE-TYPE THAT SERVE THEM

line 1:

Input-format specification (A4,6X,6I10)

line 2-?, where ? stands for the number of terminals in the network:

--Alphanumeric code for a terminal (node), here we input Z01, Z02, etc. which are short for zone 1, zone 2 ...

--Vehicle type(s) serving a stop; a "1" in column 20 means served by vehicle type 1, a "2" in column 30 means served by vehicle type 2, a "3" in column 40 means served by vehicle type 3 etc.; in our case, this is nonfunctional since only one type of vehicle is allowed.

[line ?]+1:

A blank delimiter line.

Group 4 - LINEAR YIELD-DATA

line 1:

Input-format specification (2F10.5)

line 2:

Yield-formula coefficients in terms of (yield in \$)=(intercept coeff)+(slope coeff)(distance); the example here is (yield)=0.35+(0.0)(distance), meaning a flat fare of 35 cents.

line 3:

Blank-line delimiter

Group 5 - INTER-STOP NETWORK DATA

line 1:

Input format specification (3X,I5,2I1,6I4,I5A3,I1)

line 2-?:

--[Origin(O)-node-no.]-1 specification; here the alphanumeric code of Z01 is input

line 3:

--[Destination(D)-node-no.]-1 specification, the first destination is Z02 in this example;

--O-D distance in tenths-of-a-mile (0.16 km), the distance between Z01 and Z02 is 1.8 miles in this case;

--Route-restriction code, which is 4, where 1 means operation unrestricted, nonstop service allowed, 2 means nonstop-service prohibited, 3 means operation prohibited, cannot offer single-vehicle service, 4 turnaround-service prohibited, 5 means special route-restrictions;

--maximum-no.-of-stops allowed, specified as 9;

--O-D demand, 359 passengers in this example;

--Intermediate-stop no. 1; intermediate stop no. 2 etc.; (a three-column blank between two stops means "or", while no blank between means "and"); for example, there are two intermediate-stop choices between Z01 and Z03: either stop by at Z02 or Z04;

[line ?]+1:

Origin-stop-2 specification etc., here after 6 destination nodes Z02 through Z07, origin node Z02 is now entered, and the input continues...

An echo print of the input data constitutes the first part of output. The echo print is a lot more readable and understandable than the instructions above. See RISE1.OUT for the echo print of the above set of DATA1.DAT data. In the event that a more realistic output is desirable, we have provided a text file of an echo print for an American-Airlines case-study in CASE.TXT.

E. Remote sensing and geographic information system

TS-IP USERS GUIDE

version 2.8

documented

by

Rob Urzi

for

T. S. Kelso and Yupo Chan

Updated by Yupo Chan and Brad Smith 5/97

TABLE OF CONTENTS

1. Introduction
2. Loading
3. The Program Screen
4. The Main Menu
5. Selecting Windows & Loading Images
6. Blank, Copy, and Redisplay Commands
7. The Show All & Overlay Commands
8. The Examine Command
9. Histograms
10. Zooming
11. Contour Plots
12. Palette/Color Scale
13. The Filter Menu
14. Saving an Image
15. Quitting the Program

I. INTRODUCTION

The TS-IP program is a program that loads and manipulates computer-digitized satellite images. The initials TS-IP stand for Training System\Image processing. This program allows for a variety of mathematical options that may be applied to image data, thus transforming raw spatial-data into useful information. Up to three different images may be loaded and used at the same time. The current version of TS-IP in use is version 2.8. This program was written by T. S. Kelso for the sole intention of instructional use at the Air Force Institute of Technology. Since this program was written "in-house," it continues to be revised and improved. Therefore, you may find additional options, commands and loopholes that have not been addressed in this version of the users guide. The program is copyrighted 14 Jan. 1993.

Owing to disk-space limitations, the adaptation here includes the executable code and only one image file, PENTAGON.IMG. Additional images will exceed the one-disk allowance for inclusion with this book. Obviously, the users can supplement the image file with his/her own image selections.

II. LOADING TS-IP

The TS-IP program runs under MS-DOS on any PC that has a 256 color VGA graphics-card and a VGA monitor. To begin TS-IP, you must follow these steps:

It is important to note that TS-IP only runs under DOS. To run TS-IP on a machine running a version of Windows you must open a DOS window. If the window is not already in full-screen mode you must change it to full-screen by holding the ALT key down and pressing ENTER.

From the DOS prompt, switch to the drive the software is in. For most users, this is drive C:.. Change to this drive by typing c:..

Change the directory by typing the regular change directory command `cd \book\space\images`, whichever options apply. (This will put you in the images sub-directory of the space directory).

Type the command `ts-ipx28` (This is the name of the .exe file that will run the program).

TS-IP will now load.

NOTE: If you encounter difficulties in loading TS-IP, check the settings in the file named `ts-ip.cfg` in the images directory. Make sure that the drive letter settings are correct. Also, if your video card cannot display 640x480 resolution at 256 colors, you may try to change the video mode setting. TS-IP is best run from DOS, so if you are running a Windows machine you should boot into DOS mode before running.

III. THE PROGRAM SCREEN

The first screen you will see is an informational screen about TS-IP. Hit the return key to get past this screen once you are done reading it. The next screen displayed is your program screen. This screen is divided into four sections called windows. The lower left hand window should contain a series of commands and options. This is the main menu. The other three windows, which are blank, are your working windows. It is in these three windows where the images will be loaded and displayed. Each of these windows is assigned a number. The upper left-hand corner is window number 1, the upper right-hand corner is window number 2, and the bottom right-hand corner is window number 3. The main menu window is not assigned a number because it can be accessed at all times. When commands are registered by the computer the program will signal with a beep.

IV. THE MAIN MENU

The Main Menu (in the lower left-hand corner) provides options for image manipulation. These options are presented in a format from A to Z. You need only to press the corresponding key to load that particular option. For example, if you wished to leave TS-IP, then you would press the Q key for the quit option. The following is a list of options and a brief description for each:

- A) *Add Image:* Adds an image to the image in current window.
- B) *Blank Window:* Blanks out current window.
- C) *Copy Window:* Copies the contents from one window to another.
- E) *Examine Image:* Gives either a line chart or a point chart for the x,y and z coordinates.
- F) *Filters:* Brings up a sub-menu of filtering techniques.
- G) *Group Images:* Groups multi-spectral images together.
- H) *Histograms:* Brings up a sub-menu of options for a histogram.
- I) *Invert Image:* Inverts selected image (loading upside down)
- J) *Contour Plot:* Allows for a fixed or variable plot to be drawn.
- K) *Classification:* Classifies image (histogram, 1D, 2D) Note: this function is currently *not* implemented and will cause the program to crash.
- L) *Load Image:* Loads an image into a the selected window.
- O) *Overlay:* Overlays one image on top of another. Has a sub-menu of choices.
- P) *Palette:* Changes the color scale.
- Q) *Quit:* Exits you from TS-IP, brings you back to MS-DOS.
- R) *Redisplay Image:* Redisplays the original image loaded in particular window.
- S) *Show All:* All windows display selected image, or view a very large image.
- W) *Select Window:* Selects a window for you to work on.

- X) *Cut Section:* Selects a portion of an image and writes it to disk.
- Z) *Zoom:* Magnifies a certain section of an image.
- ?) *Help Menu* Redisplays main menu in bottom left-hand corner.

Note that not every letter has an option. These options are stored for later use in future revisions of TS-IP.

V. SELECTING WINDOWS & LOADING IMAGES

Before you can load an image, you must first select one of the working windows where you would like the image to be placed. You do this by hitting the W key off the main menu (select a window). A little box will appear in the middle of your screen. This box is a visible representation of the window that is currently selected. You select the first, second or third window by either pressing the 1, 2, or 3 keys, respectively, and hit return. The illustration box will show your change and then disappear off the screen. The window you selected will remain as your working window until you change it again by using this same procedure. Images and image manipulations will remain in the selected window even if you change to another working window. A new image loaded into the same working window with an old image will replace the old image with the new one. The default working window is the number one window. Once a working window has been selected, you may load in an image by hitting the L key. A list of image data files will be displayed onto the screen. Scroll through the list with the up/down arrow keys until you find the image you wish to load. This file will be highlighted. Then hit the return key. The image will then load into your selected working window.

Example: You wish to load the file Pentagon.img into the number 3 window. First hit W to select a window and then press 3 to select the third window. Next, hit L to load the image and scroll through the list until the file Pentagon.img is highlighted. Once it is highlighted, hit the return key. The image will load onto the screen at the bottom right-hand corner.

VI. BLANK, COPY, AND REDISPLAY

The blank command will clear the selected working window. It is important to note that a working window does not have to be blanked before another image is displayed in the same space. The Redisplay command will reload the image currently loaded in that particular working window. The copy command will fill one working window with the contents of another. Therefore, the same image will be shown in multiple windows. To use the copy command, first select the working window where the image to be duplicated is located. Then select the copy command by pressing the C key and hitting return. A representation box will then appear on the screen. Press the button (1, 2, or 3) corresponding to the window where you would like to see the duplicated image located. The window does not have to be blanked; the new image will replace the old one.

Example: An image is loaded into the second working window and you would like to use it in both the second and third working windows. First, select the second window by hitting the W key and then the 2 key. Next, hit the C key for the copy command. Then press the 3 key to signify where you would like to put the duplicate image. The same image will now be located in the second and third windows.

VII. THE SHOW ALL & OVERLAY COMMANDS

Some images, such as the Pentagon.img file, are very large and would take all four windows to display. In such a case, the Show All command can fill all the working windows with the image.

Example: The pentagon.img file is actually broken down into four different sections. When it is loaded normally into the first window, you are actually viewing only the top-left corner of the entire image. You would need to use the Show All command to view the remainder of the image.

To view the remainder of an image, you first must load the image using normal procedures as discussed in part five of this document. Make sure the current selected window is the one with the desired image to view. Then press the S key for the show all command. The remaining windows will fill with the remaining parts of the image. If there are no remaining parts, the same image section will be duplicated in the other windows instead. By selecting different working windows, you are able to manipulate and view the various sections of an image.

Example: The pentagon.img file is loaded normally into the first window. To view the remainder of the image, press the S key to activate the Show All command. The top-right hand corner of the entire image will load into the second window, while the bottom right-hand corner of the image loads into the third window; thus forming one complete image. However, the bottom left-hand corner image does not fill in the fourth window due to the main menu. If you needed to view this corner, you would load the image pentagon.img file into the second window instead of the first window. Then when you select the show all command, the bottom left-hand corner would load into the third window while the first window would remain blank.

The *Overlay* command allows you to place one image over that of another to form a third image. It also allows one to view a portion of a large image that was loaded using the Show All command. The Overlay sub-menu can be accessed by pressing the O key. This sub-menu has the following options:

- 0) Average
- 1) And
- 2) Or
- 3) Xor
- 4) Maximum
- 5) Minimum
- 6) Difference
- 7) Subtract
- 8) Dis Last (display last)
- 9) Dis All (display all)

The desired option can be chosen by pressing the corresponding numeric key. The program will then ask you which image file that you would like to use. Select the window where you would like to overlay the image.

VIII. EXAMINE IMAGE

The Examine Image command allows you to view an image by its x, y and z (gray shading) coordinates for any given pixel (point). It also allows you to examine the image by a vertical line-by-line representation of the peaks and valleys. To run the examine command, first select the window with the desired image. Then hit the E key. A small sub menu will appear and list out two options: either 0) Point or 1) Line. Hit the 0 key if you wish to examine the xyz coordinate or 1 if you wish to examine by using the "peaks and valleys" line. If you hit the 0 key, a small cross-hair arrow will appear over the image. You may move this arrow to any point over the image by using the arrow keys. A corner box will appear on the monitor and display the x, y, z value of the current spot. When you are done examining the image press the escape (Esc) key.

By pressing the second option a vertical line will be drawn across the face of the image. A vertical line bar will also appear in the window over the main menu. A second line will be drawn over the window where the main menu is located. This is the "peaks and valleys" line representing the data points

along the line that was placed over the image. The line over the image can be moved from left and right with the arrow keys. Once your examination is done press the escape (Esc) key. Then hit the ? key to bring back the main menu.

Example: You wish to examine the xyz coordinates of an image that is located in the third working window. After you select the third window hit the E key. Then press the 0 key. A cross-hair appears in the image along with the coordinate box. Note again that the z coordinate is the gray shading value, where the lower the number the darker the image. Once you are done examining the image hit the Esc key.

IX. HISTOGRAMS

You may display a histogram of the image by pressing the H key. You will be presented with a sub-menu of four options:

- 0. Display
- 1. Linstretch
- 2. Logstretch
- 3. Equalize

Press the corresponding numeric key for the desired choice. The histogram will appear over the window with the main menu. Do not be alarmed if it takes several minutes for it to load. When you are finished hit the escape key and the ? key. The main menu will then appear.

Example: You wish to examine the histogram of an image using the linear stretch option. After you select the appropriate window hit the H key. Then press the 1 key. The histogram will appear with a diagonal line through the middle and a vertical line at the left edge of the box. Using the arrow keys, move the vertical line to the position you want it and press enter. A vertical line will then appear at the right edge of the box. Again, move this to the correct position and press enter. The pixel values remaining between the vertical bars will then be stretched out linearly among the entire palette, and the image will be redrawn. Once you are done examining the image hit the Esc key.

NOTE: This might take several minutes (it acts as if the system crashed but that is not the case- it just takes a while to load back in).

X. ZOOMING

The zoom command allows you to magnify a certain area of an image. To use the zoom command, first select the window with the desired image, and then press the Z key. Four zooming options will appear:

- 0. Blow Up
- 1. Neighbor
- 2. Bilinear
- 3. Cubic

Press the numeric key for the desired choice. A square will appear over the image. Position the square over the area of the image in which you would like to magnify by using the arrow keys. Once it's in position, hit the return key. The window selection box will now appear; select the window in which you would like to place the magnified image. The magnified image will now appear in the desired window.

Example: The image Pentagon.img is loaded in the third window, and you wish to zoom in on the right-most corner and place magnification in the second window. First, you select the third window and press the Z key. Next, position the box in the corner by moving it with the arrow keys. Then press the return

key and select the second window by hitting the 2 key. The magnification will now appear in the second window.

Note: The cut section command is almost identical to the zoom command in operation but is different in that it cuts out a certain area rather than magnifying it.

XI. CONTOUR PLOTS

TS-IP allows you to examine a contour plot of an image. To use this command, first select the window with the desired image to contour. Then press the J key. A sub menu of choice will appear. The plot may be loaded as either being fixed (1 through 5) or variable (1-5). Choose the desired option by pressing the corresponding numeric key.

- 0. Fixed 1
- 1. Fixed 2
- 2. Fixed 3
- 3. Fixed 4
- 4. Fixed 5
- 5. Variable 1
- 6. Variable 2
- 7. Variable 3
- 8. Variable 4
- 9. Variable 5

These numbers represent the number of gradations which will be shown. Once you have made your choice, the program will prompt display the gradations in the fourth window along with the shading graph. Press the return key to cycle through these lines. Next, the program will ask you for the window in which to display this plot. Select this desired window by pressing either 1, 2 or 3. and hitting return. When you are finished with your examination press the escape key and then ? to return the main menu.

Example: You wish to view the contour plot using 3 fixed gradation values of the pentagon.img image, which is currently loaded onto window one. Select the first window and then hit the J key, then press the 2 key. The line gradation plot will appear over the main menu. Press the return key three times to view the three different separation lines. Then select the desired output window by pressing either 1, 2 or 3. After you are done, return to the main menu by pressing the Esc key followed by the ? key.

XII. PALETTE/COLOR SCALE

The color scale may be changed depending on the image clarification and/or personal taste. This scale is based on principle of either shading from light to dark or using a color spectrum. The program defaults on a 256 gray-shading scale. The color scale may be changed by pressing the P key, which will give you a list of 30 various color schemes from which to choose from. You may circulate through this list by using the arrow keys. Press return when you highlight a scheme of your choosing. The scheme may be changed as often as you desire.

Example: You wish to change the 256 Gray shading scheme to a color rainbow scheme. Therefore, press the P key and cycle through the list with the arrow keys until the choice Rainbow is highlighted. Hit the return key, and the screen will instantly change to the new scheme.

XIII. THE FILTER MENU

Filtering is one of the most important parts of image manipulation. Various commands such as Sobel convolution, fast Fourier transform, Laplacian transformations, and noise removal are just some examples of how an image can be clarified. The Filter option will bring up a sub menu of choices which will perform these commands. The sub-menu is presented in a format identical to that of the main menu. The options off this menu are:

- A) Average
- B) Blur
- C) Crispen
- D) Line Detection
- F) FFT
- L) Laplacian
- M) Median
- N) Noise Removal
- S) Sobel
- X) Sharpen
- Z) Special

The options are chosen by pressing the appropriate letter key. Many of these options will give you a sub-menu of choices. Average & Median (options A & M) will give you a 3x3 and a 4x4 option. Crispen and Sharpen will give you a high, medium, and low option. Line detection can be horizontal, vertical, diagonal (top-left to bottom right) or diagonal (top right to bottom left). Noise removal can be primary, secondary, 3x3, or 4x4. Sobel convolution will present a sub-menu of G(x) and G(y) options. Regardless of which of these commands that you give the program, it will prompt you for the destination display window. Once the filtered image is displayed, the main menu will reappear in the bottom-left corner.

Example: You wish to filter an image using the 3x3 noise removal algorithm. Therefore, you would select the appropriate window and press the F key to bring up the Filters menu. Then press the N key to select Noise Removal, and choose 2 for Median 3x3. The program will process the image from left to right as you watch. You can then continue to work with this window.

XIV. SAVING AN IMAGE

There are two ways to save an image. You may either save the window or save the entire screen. To save a window first highlight the appropriate window using the 'W' command and then press the comma key. To save the screen simply press the period key at any time. These commands create several files in the \book\space\images\work subdirectory. For example, if you saved a window and there were no previous saved files in the directory, the files created would be named window-a.raw and window-a.bat. Subsequent files would be named window-b.raw, window-c.raw, and so on. Screen saved files are named screen-a.raw and screen-a.bat, for example. The raw format is not used by most software, but by using these two files it is possible to create a gif file, which is a common image format. After you exit TS-IP change directories to the work subdirectory by typing `cd work`. Then execute the batch file corresponding to the name of the image you saved. In the above example, you would type `window-a` which would create a file named `window-a.gif`.

XV. QUITTING THE PROGRAM

You may leave TS-IP by selecting the quit option off of the main menu (pressing the Q key). This will bring you back to DOS at the following location:

```
c:\book\space\images>
```

You can exit the TS-IP program at any time by pressing down the control key and hitting the break key (i.e. Ctrl-Break). This is useful if the program locks up or crashes.

F. Multiple travelling-salesmen-problem

Welcome to the **SPACE FILLING CURVE** (SFC) Heuristic Program! This program was originally coded by W. B. Carter (1990), and the document is distributed by the Defense Technical Information Center (DTIC). SFC is primarily a heuristic vehicle-routing algorithm as described in the "Measuring spatial separation" chapter of this book.

This directory holds the capability to accomplish 2- and 3-dimensional space-filling-curves.

The contents of this directory consist of the following:

- 1) GWBASIC is the executable file for compiling a source BASIC program.
- 2) SFC.BAS is a basic program performing the space-filling-curve calculations.
- 3) SFC.BAT is a DOS batch-file that starts GWBASIC, loading, compiling and running SFC.BAS.
- 4) Several sample text-files (delimited) are also included, containing data and operating instructions.

HOW TO USE THIS DISK:

I. Put disk in drive A: and change to drive A: (unless you copy the contents of the book disk to a hard-drive directory)

II. At the A:\prompt, type "sfc"

III. Now just answer the questions! (At any time, one can exit by pressing 'CTRL' and c, and then typing "system" to exit GWBASIC. A short text-file that explains the most basic of GWBASIC commands can be located under the BASIC.TXT file at the root directory)

The files named after major U.S. cities are good examples of how input data-files must be set up.

EXAMPLE: (BOSTON)

```
"ALBANY",314,37
"BOSTON",331,34
"NORTHAMPTON",325,37
```

This program was originally designed for use with global (latitude-longitude) positions converted to grid positions. The first value must be a name for its associated geographic-location and must be enclosed in quotation marks " ". The next two values are the x and y coordinates (navigational grid values for the example). As with any delimited file, you must include the commas to separate the inputs. The space-filling-curve heuristic uses these last two values to transform a two- or three-dimensional space into the unit interval (for 3-dimensional problems the program will prompt for the third dimension for each geographic location while it is executing).

For example, say you have points 1,2, and 3. Their x-y coordinates are (1,1), (2,2) and (3,3) respectively. The input file should read:

```
"1",1,1
"2",2,2
"3",3,3
```

In a 2-dimensional calculation, the theta values (the unit-interval or Sierpinski-curve values) will be produced normally. For a 3-dimensional

problem the program will ask for the z coordinate for each geographic location, as mentioned above.

Another value the program prompts for is the "number of binary digits." This value tells the program how many times to divide the "unit square or cube" for mapping to the unit interval. The object here is to use the smallest value possible that will return a unique theta value for each point. This holds true for both the 2- and 3-dimensional cases. Three-dimensional cases are tricky. The best way to obtain answers in 3-dimensional cases is to start with a "binary digit" value of 1, then try 2, then try 3, etc. The program runs fast, so this does not take as much time as one would think. (NOTE: The 3-dimensional cases cause the arrays to grow rapidly, eating up memory! This program has returned good results for up to 52 3-dimensional points. Above this number, the 3-dimensional capability is unknown! The 2-dimensional capability has been tested to 89 2-dimensional points.)

GOOD LUCK and have fun! The Space Filling Curve is an amazingly versatile tool!

G. A location-routing delivery-model

This program was originally coded as part of Baker (1991). The document is distributed by the U.S. Defense Information Technical Center (DTIC) and explained in the "Location-routing" chapter of this book. The study extends work done by the Military Airlift Command (now the Air Mobility Command) Analysis Group on reducing the operating costs of the Defense-Courier-Service network.

The study's primary focus is to minimize those costs by varying the number and location of servicing depots, and the routes flown from those depots. The theoretical algorithm used in the methodology is an expansion of Laporte et al.'s (1986) formulation of the multiple-depot multiple-traveling-salesman facility-location problem (which is described in the "Measuring spatial separation" chapter of this book.) Multiple servicing-frequency is addressed by clustering co-located demands with Kulkarni and Bhavne's (1985) subtour-breaking constraint. Vehicle range is considered by redressing a shortfall of the subtour-breaking constraint, which was first noted by Brodie (1988). The formulation is merely used as a *validation* of a system-wide solution heuristic, since exact solution is beyond the range of current computing capabilities.

The solution heuristic is a combination of the *minimum spanning-forest* (Prim and Dijkstra) and the *Clarke-Wright method*. The spanning-forest is used for depot location and partitioning of the study area into service regions, while the Clarke-Wright computes the routes flown from the depots to their assigned service points. The heuristic is suboptimal by 3.8% on the average in six validation runs, with no run greater than 20%. The results indicate several depots may be closed without large increase to system mileage. The entire case study is documented in the "Simultaneous location-routing" chapter of this book.

The software contained here is organized under the **SPANFRST** directory. Sample data files are included to illustrate program execution.

SPANFRST DIRECTORY CONTENTS:

The logic of this program is explained in the book under the "Location-Routing" chapter in which Baker's combined location-and-routing model is fully explained. This master directory contains three subdirectories.

1. The subdirectory labeled *VALID* provides the most comprehensive overview of the various files, including other subdirectories. It contains all the files used in *model validation* using the extended Laporte formulation, as mentioned above and fully described in the "location-routing" Chapter of the accompanying book. Following standard practice, there is a *readme.txt* file within the *VALID* sub-directory which should also be referred to.
2. The subdirectory labeled *MSF* contains the FORTRAN file for *minimum-spanning-forest* as mentioned above and also described in the "Location-routing" Chapter of the accompanying book. Since the same code was used repeatedly, only one sample input and output file accompanies the code file. Refer to Baker (1991) Appendix E for a comprehensive listing of various output that can be generated.
3. The subdirectory labeled *CW* contains the FORTRAN file for the *Clarke-Wright heuristic*. The generic algorithm is described in the "Measuring Spatial Separation" Chapter and the special adaptation here is documented in the "Location-routing" chapter of the book. As before, only sample input and output files are included, since the full output is given in Appendix E of Baker (1991).

The reader can execute any of these subdirectories independently even though they perform an integrated task when they are all invoked. For an

introduction, the reader is referred to the VALID subdirectory. To the extent that this sub-directory contains codes and data that validate heuristic against analytic models, it provides the most comprehensive overview of the entire software package.

H. Derivation-allocation model for state parks

This program was originally developed at the State University of New York at Stony Brook by T. O. Carroll, Y. Chan, R. Silkman, and T. Sexton (1982). The study is documented in Chan and Carroll (1985). Subsequent extensions were performed at Washington State University and the Air Force Institute of Technology by Y. Chan and his associates. This program, **STATEPRK**, generates demand for state-park visitation and allocates the demand among competing state parks. The model is described in the "Activity generation and allocation" chapter of this book.

This disk contains the following files under STATEPRK:

PARKNY.ASC : BASIC source-program containing the New-York-Parks data written to run under the GWBASIC compiler.

NYPARKS.BAT : DOS batch file that initiates GWBASIC, which compiles and executes PARKNY.ASC.

PARKWAS.ASC : BASIC source-program containing the Washington-State-Parks data written to run under GWBASIC.

WASPRKS.BAT : DOS batch-file that initiates GWBASIC, which compiles and executes PARKWAS.ASC.

GWBASIC.EXE : BASIC-program compiler that turns PARKNY.ASC and PARKWAS.ASC into executable codes. A short text-file that explains the most basic of GWBASIC commands is contained in the file BASIC.TXT at the root directory.

Additional information:

1. The model (and particularly the input/output data) behind the New-York-Parks Program is explained in the book under the "Activity generation-distribution" chapter, as pointed out above. Data for the Washington-Parks program is similar.
2. To run the New-York-Parks Program, type in "NYPARKS".
3. To run the Washington State Parks Program, type in "WASPRKS".
4. Being developed in a university setting, the codes are written in a prototype style. In order to run a new dataset, one needs to go into the source program and change the "DATA statements."

I. Garin-Lowry model

This software represents a state-of-the-art implementation of the Garin-Lowry family of land-use models. Instead of the iterative procedure used in the original Lowry-model, this code provides a much more satisfactory way to performed constrained forecasts. The code originated in Yi and Chan (1988). Subsequent extension was made in the Air Force Institute of Technology by Brad Smith and Yupo Chan. Complete documentation of these programs can be found in the "Chaos, catastrophe, bifurcation and disaggregation" chapter in the book.

This directory contains the following files:

AFTUCAL.FOR: source code to calculate disaggregate multipliers a and f and basic accessibility functions t and u
AFTUCAL.EXE: executable code for AFTUCAL.FOR
FIB.FOR: source code to forecast target-year population using a Fibonacci-search technique and to calculate elasticity parameters b and c. The default is to run the "do-nothing" alternative, but if a flag is set in the data file, the "do-something" alternative may be run.
FIB.EXE: executable code for FIB.FOR
25MPH.FOR: source code for program with constraints for aggregate and disaggregate case with "do-something" alternative, a 25-mph personal-rapid-transit (PRT) system
25MPH.EXE: executable code for 25MPH.FOR
DONOTHIN.FOR: source code for program with constraints for aggregate and disaggregate case with "do-nothing" alternative
DONOTHIN.EXE: executable code for DONOTHIN.FOR
LMBI42.DAT: data file for 42-region case-study for AFTUCAL.EXE, FIB.EXE, 25MPH.EXE, and DONOTHIN.EXE
LMBI4.DAT: data file for 4-region example in book for AFTUCAL.EXE, FIB.EXE, 25MPH.EXE, and DONOTHIN.EXE
README.TXT: this current file

To run a program simply type the name of the executable (the .exe file). The Program will prompt you to enter the name of a data file and the name of an Output file. If you enter the name of an output file that already exists the old file will be overwritten. The data file must be in the same format as the sample data file that comes with each program. The format of each data file and the variables it contains are detailed at the end of the sample data file. For example, if you want to run AFTUCAL.EXE, type AFTUCAL at the DOS prompt. The program will ask you to enter the name of the data file. If you want to run the 4-region example from the book, type LMBI4.DAT. If you want to run the 42-region case-study, type LMBI42.DAT. The program will then ask you to enter the name of an output file. Type in AFTU4.OUT, for example. The program will notify you when it has stopped and return you to the DOS prompt. If the programs are run on the LMBI4.DAT file they should finish almost immediately, but the LMBI42.DAT file may take up to several minutes to complete.

The Fibonacci program runs by default on the "do-nothing" travel-time-matrix data. If you want Fibonacci to run on the "25 mph PRT" travel-time matrix, simply edit the data file and set the "PRT" flag to one.

II. References

- Achutan, N. R.; Caccetta, L. (1991). "Integer linear programming formulation for a vehicle routing problem" *European Journal of Operational Research*, Vol. 52, pp. 86-89.
- Andersson, A. E.; Kuenne, R. E. (1986) "Regional economic dynamics" in *Handbook of Regional and Urban Economics*, Vol I, Edited by P Nijkamp, Elsevier.
- Araque, J. R. (1989) "Contributions to the polyhedral approach to the vehicle routing problem" PhD dissertation, State University of New York at Stony Brook, Stony Brook, New York.
- Baker, S. F. (1991) "Location and Routing of the Defense Courier Service Aerial Network" Master's thesis, AFIT/GOR/ENS/91M-1, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio (Advisor: Y Chan).
- Banaszak, D.; Cordeiro, J.; Chan, Y. (1997) "Using the K-medoid and Covering Approaches in Pattern-Recognition Problems" Working paper, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio.
- Carroll, T. O. ; Chan, Y.; Silkman, R. and Sexton, T. (1982) "An Assessment of Regional and State-wide Economic Impacts for New York State Parks" Institute for Urban Science Research, State University of New York at Stony Brook, Stony Brook, New York.
- Carter, W. B. (1990) "Allocation and Routing of CRAF MD80 Aircraft" Master's thesis, AFIT/GST/ENS/90M-4, Department of Operational Sciences, Air Force Institute of Technology, Wright-Patterson AFB, Ohio (Advisor: Y Chan)
- Chan, Y. (1974) "Configuring a transportation route network via the method of successive approximations" *Computers and Operations Research*, Vol. 1, pp. 385-420.
- Chan, Y. (1979) "A graph theoretic method to quantify the airline route authority" *Transportation*, Vol. 19, No. 1, pp. 309-320.
- Chan and Carroll (1985) "Estimating recreational travel and economic values of state parks" *Journal of Urban Planning and Development*, Vol. 111, pp. 65-79, American Society of Civil Engineers.
- Chan, Y.; Rasmussen, W. (1979) "Forecasting housing requirements in a college town" *Journal of the Urban Planning and Development Division*, Vol 105, pp 9-23.
- Cliff, A. D.; Ord, J. K. (1981) *Spatial Processes: Models and Applications*, Pion.
- Chrissis, J. W. (1980) "The Solution of Nonlinear Pseudo-Boolean Optimization Problems Subject to Linear Constraints" PhD dissertation, Department of Industrial Engineering and Operations Research, Virginia Tech, Blacksburg, Virginia.
- Doreian, P. (1981) "On the estimation of linear models with spatially distributed data" Chapter 11 in *Sociological Methodology*, editor S S Lienhardt, The American Sociological Association, School of Urban and Public Affairs, Carnegie-Mellon University, Pittsburgh, Pennsylvania.
- Fullingham, D.; Chan Y. (1997) "Prepositioning and repositioning" Working paper, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio.

- Hansen, W. G. (1959) "How accessibility shapes land use" *Journal of the American Institute of Planners*, pp 152-166.
- Irish, T.; May, T.; Chan, Y. (1995) "A Stochastic Facility Relocation Problem" Working paper, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio.
- Jha, K. (1972) "Demographic models" Working paper, Department of Civil Engineering, Pennsylvania State University, University Park, Pennsylvania.
- Junio, D. F. (1994) "Development of an Analytic Hierarchy Process for Siting of Municipal Solid Waste Facilities" Master's Thesis, Dept. of Engineering and Environmental Management, Air Force Institute of Technology, Wright-Patterson AFB, Ohio (Advisor: Y Chan).
- Kerschus, C.; Ee, M.; Chan, Y. (1996) "Location determination of obnoxious facilities where two major concerns are examined: Seismic risks and public opposition" Working paper, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio.
- Kulkarni, R. V.; Bhavne, P. R. (1985) "Integer programming formulations of vehicle routing problems" *European Journal of Operational Research*, Vol 34, pp 403-404.
- Laporte et al. (1986) "An exact algorithm for solving a capacitated location-routing problems" *Annals of Operations Research*, Vol 6, pp 293-310.
- Mehrez, A.; Eben-Chaime, M.; Brimberg, J. (1996) "Locational analyses of military intelligence ground facilities" *Military Operations Research*, Vol 2, No. 3, pp 61-69.
- Nagurney, A. (1993) *Network Economics: A Variational Inequality Approach*, Kluwer Academic Publishers.
- Neddef, D. (1994) "A remark on 'Integer linear programming formulation for a vehicle routing problem' by N R Achutan and L Caccetta, or how to use the Clarke and Wright Savings to write such integer linear programming formulations" *European Journal of Operational Research*, Vol 75, pp 238-241.
- Pak, P. S.; Kim, G.; Suzuki, Y. (1989). "Multizonal model of industrial activities based on journey time model", Working Paper, Faculty of Engineering, Osaka University, Yamada-oka 2-1, Suita, Osaka 565, Japan.
- Patterson, T. S. (1995) "Dynamic Maintenance Scheduling for a Stochastic Telecommunication Network: Determination of Performance Factors" Master's Thesis, Department of Operational Sciences, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio. (Advisor: Y. Chan)
- Patterson, K.; Horton, K. G.; Chan, Y. (1994) "Games with multiple payoffs" Working Paper, Department of Operational Sciences, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.
- Putman, S. (1972) "A 'workbook' for education and limited research use of the Lowry model," Department of City and Regional Planning, University of Pennsylvania, Philadelphia, Pennsylvania.
- Steppe J. M. (1991) "Locating Direction Finders in a Generalized Search and Rescue Network" Master's Thesis, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio (Advisor: Y Chan).
- Suzuki, Y.; Pak, P. S.; Kim, G. (1989). "Impact analyses of construction of Kansai International Airport", *Journal of Urban Planning and Development*, American Society of Civil Engineers, Vol. 115, No. 1, pp. 33-49.

Tapiero, C. S. (1971) "Transportation-location-allocation problems over time" *Journal of Regional Science*, Vol 11, pp 377-384.

Weir, J.; Stewart, J.; Chan, Y. (1994) "Single facility, multiple tour allocation problem" Working paper, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio.

Wright, S. (1995) "Spatial Time-series: Pollution Pattern Recognition under Irregular Interventions" Master's Thesis, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio (Thesis advisor: Y Chan).

Wright, S.; Chan, Y. (1994) "Pure and Polluted Ground Water Classification on a Pixel Map" Working paper, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio.

Wright, S.; Chan, Y. (1994a) "MCDM Applied to the ICM Contextual Image Classification Technique" Working paper, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio.

Yi, P. (1986) "Infrastructure Management: A Bifurcation Model in Urban/Regional Planning" Master's Thesis, Department of Civil & Environmental Engineering, Washington State University, Pullman, Washington (Advisor: Y Chan).

Yi, P.; Chan, Y. (1988) "Bifurcation and disaggregation in Lowry-Garin models: Theory, calibration and case study" *Environment and Planning A*, Vol. 20, pp. 1253-1267.

Zelany, M. (1982) *Multiple Criteria Decision Making*, McGraw-Hill.

[reference.yc]