LOG TRUCK SCHEDULING
BY NETWORK PROGRAMMING

by

Zhenyan Shen

A PAPER

submitted to

The Department of Forest Engineering

in partial fulfillment of
the requirements for the
degree of

Master of Forestry

August 18, 1988

AN ABSTRACT OF THE PAPER OF

Zhenyan Shen for the degree of MASTER OF FORESTRY in FOREST ENGINEERING presented on August 18, 1978.

Title: Log truck scheduling by network programming

Abstract approved: ______________________________

Dr. John Sessions

In northeast China, logging has three stages; timber harvest, transportation and the operations inside a timber yard. The timber yard is an intermediate transhipment point between truck transport and rail transport to manufacturing centers. The transhipment capacity of the timber yard is often the limiting acitivity in the logging chain.

Since the transportation is the middle part, its management affects both the technology and the economics of the other two. At the same time, the other two stages provide constraints to the transportation operations.

The objective of this paper is to find an optimal economic plan for assigning trucks to timber yards in such a way as to minimize the number of trucks while satisfying the delivery schedule for the timber yard. This is done by controlling the log truck arrival distribution over time. As a result, the fewest number of trucks are used and the timber yard is efficiently used.

A computer program was developed to provide the transportation manager with detailed information about truck

allocations and to permit rapid updating of the transportation plan.

The truck allocation problem is formulated as a capacitated network problem and the out-of-kilter algorithm is used to solve it. After the network problem is solved, the number of trucks needed to carry out the schedule can be determined through an hour-by-hour truck inventory analysis.

LOG TRUCK SCHEDULING
BY NETWORK PROGRAMMING

by

Zhenyan Shen

A PAPER

submitted to

The Department of Forest Engineering

in partial fulfillment of
the requirements for the
degree of

Master of Forestry

August 18, 1988

APPROVED:

____
Professor of Forest Engineering in charge of major

____
Head of Department of Forest Engineering

Date thesis is presented August 1988

Typed by and for Zhenyan Shen

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

1. PROBLEM DESCRIPTION

In the Chinese forest industry, large distances separate the places where timber is produced and the manufacturing centers where it is used. Railroad transport is used for long distance transport and trucks are used for short distance transport from the landings to the railroad. A timber yard is constructed at the transhipment point between the two transportation modes. Large truck transportation companies under a forestry bureau are responsible for tree length transportation from the landings to the timber yard. The parking lots with a pool of log trucks of those transportation companies are located at the same towns as the timber yards. This is true for almost all the forestry bureaus.

After the log trucks arrive at the timber yard, trucks are unloaded by crane and the tree lengths are bucked. The logs then are rolled on to belt transport machines, and rolled off the belt transportation machines at different decks according to the species of the log and their sizes. After sorting, the logs are decked and are ready to be loaded onto trains and transported to different areas of the country.

Therefore, the operating procedure inside the timber yard is (1) unloading; (2) bucking; (3) sorting; (4) decking and (5) loading. Step (1) through step (4) can be viewed as an assembly line. The flow (volume per hour or truck loads per hour) of timber through each step is the same. Therefore, the maximum of the flow through this line is restricted to equal the minimum flow of all the steps.

If the flow of the timber, consequently the number of log trucks arriving at timber yard per hour, is equal to maximum flow of the operation line, the line will work at maximum efficiency. On the other hand, if the flow is larger than the maximum flow of the operation line, the surplus part of the flow can not get into the line. When this happens, a queue of unloaded trucks begins. The waiting time can not be controlled by the transportation company and this may aggravate further cycles.

Table 1 and Table 2 show data collected at the Dong Jing Chen Forestry Bureau and Mu ling Forestry Bureau in northeast of China in 1981. Numbers in the tables are the number of trucks which arrived at the timber yard in each time interval. The number of trucks which can be unloaded per hour is estimated to be 4 trucks approximately for Dong Jing Chen forestry bureau and 5 for Mu Ling Forestry Bureau. The sum of waiting time in hours through each day is presented in the right column. The total waiting time in about fifty days can be found at the bottom of tables. The data suggests the in line waiting time can be very long if the arrival distribution of the log trucks fluctuates too much over time.

Tabl 1. TRUCK ARRIVAL AT TIMBER YARD.

(Dong Jing Chen Forestry Bureau)

DATE	TIME INTERVALS																									WAITING TIME
	[18-19]	[19-20]	[20-21]	[21-22]	[22-23]	[23-24]	[24-1]	[1-2]	[2-3]	[3-4]	[4-5]	[5-6]	[6-7]	[7-8]	[8-9]	[9-10]	[10-11]	[11-12]	[12-13]	[13-14]	[14-15]	[15-16]	[16-17]	[17-18]	
JAN. 1	3	2	3	0	4	3	2	3	3	3	0	2	1	1	3	2	1	0	3	2	2	6	5	2	6
JAN. 2	0	2	3	3	1	3	2	0	3	3	2	2	1	0	4	1	1	0	1	1	4	1	4	3	0
JAN. 3	2	4	6	0	2	1	3	0	3	10	4	5	0	0	2	3	2	6	2	3	0	2	2	2	26
JAN. 4	0	7	1	3	2	3	3	2	4	3	0	3	3	1	1	4	0	2	5	4	4	1	1	3	6
JAN. 5	0	4	1	0	6	5	0	0	4	1	4	2	3	2	4	0	1	1	4	2	2	3	2	2	5
JAN. 6	2	2	2	3	5	3	2	2	5	2	3	0	3	2	4	1	1	2	1	4	4	4	2	2	2
JAN. 7	4	2	0	2	1	1	0	2	4	1	0	4	0	4	0	2	2	3	3	6	2	3	1	2	2
JAN. 8	1	1	3	5	2	1	1	3	5	1	3	2	2	3	1	5	4	0	1	4	4	3	2	3	4
JAN. 9	0	3	2	1	4	1	1	9	4	3	2	2	3	4	0	3	2	2	2	4	2	6	3	2	19
JAN.10	4	3	1	0	4	2	3	3	5	4	2	1	2	4	3	3	1	3	1	7	4	3	0	1	10
JAN.11	3	2	4	1	1	2	4	5	3	3	1	2	4	2	3	1	3	1	3	7	0	2	4	5	5
JAN.12	3	2	4	4	0	2	1	0	0	2	5	2	5	2	2	1	4	1	7	4	3	1	5	1	11
JAN.13	0	2	4	0	0	7	3	4	3	3	2	4	1	3	1	1	4	2	2	1	5	3	3	3	9
JAN.14	1	4	3	2	2	1	6	1	3	0	4	0	3	7	3	1	3	0	3	4	0	1	5	0	8
JAN.15	4	5	2	6	1	3	4	4	4	1	2	2	1	1	4	3	1	1	5	0	4	8	3	4	14
JAN.16	1	3	3	2	2	1	0	1	2	1	4	5	2	2	1	0	4	2	9	7	5	3	3	3	44
JAN.17	2	4	3	0	1	1	8	2	3	3	3	3	1	1	4	0	6	2	1	4	2	1	3	3	20
JAN.18	0	3	4	5	0	2	4	1	5	1	3	3	2	3	1	4	2	2	2	4	1	1	6	5	7
JAN.19	1	4	1	3	5	2	1	2	3	1	4	3	2	3	1	3	1	0	5	2	4	0	3	8	6
JAN.20	0	0	3	5	1	5	1	2	3	3	1	3	1	6	2	1	3	4	4	2	3	0	4	3	4
JAN.21	3	2	0	4	1	1	1	4	3	3	5	1	6	1	2	2	4	2	0	3	2	1	3	5	4
JAN.22	1	4	1	2	4	1	5	0	2	3	2	1	4	3	5	1	3	2	3	2	3	3	4	1	2
JAN.23	2	1	0	2	6	3	2	5	1	4	0	1	2	6	2	3	2	1	4	2	1	3	1	4	6
JAN.24	5	1	0	6	4	0	4	1	2	1	0	3	3	2	1	1	1	0	4	5	3	2	9	5	17
JAN.25	1	4	4	2	1	0	1	2	2	3	2	2	1	3	0	1	2	2	1	2	4	3	1	0	10
JAN.26	1	1	6	4	2	2	1	4	3	1	2	6	1	2	1	0	3	0	0	2	2	2	6	4	10
JAN.27	4	2	3	5	5	5	1	4	0	6	3	3	1	5	2	4	0	2	1	3	2	4	2	2	12
JAN.28	1	5	3	8	3	2	2	2	1	6	2	2	3	3	2	0	1	1	3	2	3	1	1	4	6
JAN.29	10	1	1	6	4	2	3	2	5	2	3	3	6	2	1	3	1	0	3	3	2	4	1	0	16
JAN.30	8	3	2	3	3	2	6	0	5	4	0	2	2	3	0	2	1	1	1	4	1	1	3	5	13
JAN.31	5	1	1	1	4	4	2	1	1	1	2	3	5	2	2	1	0	0	2	1	1	1	2	5	4
FEB. 1	2	5	3	1	3	4	2	2	2	1	2	2	1	2	0	0	1	2	3	3	6	4	3	0	6
FEB. 2	2	2	2	1	1	8	1	2	3	1	2	2	6	4	1	1	2	2	3	3	1	5	1	4	10
FEB. 3	4	7	2	2	2	1	6	3	2	1	2	4	0	4	0	4	2	0	2	3	3	5	0	4	8
FEB. 4	1	6	2	1	0	1	1	1	4	1	0	2	4	2	1	1	1	0	2	2	2	3	6	3	4
FEB. 5	1	2	2	1	2	1	3	1	1	0	0	1	0	2	1	1	0	1	1	3	2	2	5	3	1
FEB. 6	1	3	2	0	3	0	0	1	1	1	1	0	1	0	0	0	0	1	4	4	3	1	1	0	0
FEB. 7	3	5	2	3	2	1	1	0	0	2	0	0	0	0	1	1	0	1	4	6	0	4	4	6	5
FEB. 8	2	3	2	0	3	3	2	1	1	0	0	0	0	0	0	0	0	4	3	3	3	2	6	3	3
FEB. 9	4	3	3	5	1	3	3	1	2	0	0	3	0	1	1	2	1	0	1	6	1	5	2	3	5
FEB.10	4	5	3	5	0	3	3	2	2	2	1	0	1	1	2	1	0	0	3	4	2	3	3	3	2
FEB.11	5	3	3	7	3	5	1	1	2	0	2	3	1	1	2	1	0	2	2	1	2	3	1	5	10
FEB.12	0	1	10	3	2	2	2	2	3	5	2	0	4	2	1	2	2	1	2	3	1	7	7	2	29
FEB.13	3	4	5	4	3	2	1	2	5	3	2	0	5	1	1	3	0	4	0	1	1	4	0	5	21
FEB.14	4	2	2	3	2	3	5	3	1	1	2	3	3	1	1	1	3	1	1	5	5	1	5	7	10
FEB.15	5	5	4	2	3	5	4	1	4	1	2	0	5	1	4	1	2	2	4	2	2	5	1	4	36
FEB.16	4	7	1	3	2	4	5	4	1	5	1	1	2	2	2	2	1	1	4	3	4	2	2	4	6
FEB.17	2	2	3	1	1	2	5	1	5	3	5	5	6	1	0	0	4	1	2	5	1	0	2	4	11
FEB.18	2	3	6	2	6	3	0	3	6	4	1	4	2	4	0	1	0	1	2	1	3	2	3	4	9
FEB.19	3	5	7	5	2	5	0	1	2	5	0	3	4	1	3	0	0	0	5	2	3	3	6	6	31
FEB.20	4	6	3	1	0	0	3	3	3	5	4	0	3	0	0	7	1	0	2	1	2	4	1	8	26

TOTAL IN LINE WAITING TIME = 541 HOURS

Tabl 2. TRUCK ARRIVAL AT TIMBER YARD.

(Mu Ling Forestry Bureau)

DATE	TIME INTERVALS																								WAITING TIME
	[1-2]	[2-3]	[3-4]	[4-5]	[5-6]	[6-7]	[7-8]	[8-9]	[9-10]	[10-11]	[11-12]	[12-13]	[13-14]	[14-15]	[15-16]	[16-17]	[17-18]	[18-19]	[19-20]	[20-21]	[21-22]	[22-23]	[23-24]	[24-1]	
JAN. 1	6	2	5	2	1	2	2	0	0	1	0	0	4	4	3	6	8	4	3	7	5	5	7	8	32
JAN. 2	6	3	2	3	1	0	0	0	0	0	1	2	3	3	3	2	4	4	7	8	4	11	10	2	70
JAN. 3	4	8	3	0	2	2	0	0	0	0	0	0	2	3	6	4	7	7	2	4	10	3	4	2	67
JAN. 4	3	7	3	2	1	0	3	0	0	0	0	3	4	3	7	8	5	7	5	3	4	3	13	4	58
JAN. 5	5	8	3	1	2	0	0	0	1	0	1	1	1	5	1	10	5	7	4	9	5	9	3	10	126
JAN. 6	2	1	3	1	7	4	0	0	0	0	0	3	3	5	4	4	7	4	5	5	5	11	2	8	71
JAN. 7	3	1	7	8	1	3	0	0	0	0	0	1	3	6	2	6	6	3	3	8	7	4	6	8	46
JAN. 8	5	3	8	4	1	1	0	0	2	0	0	1	5	4	4	6	2	3	3	5	10	4	5	4	52
JAN. 9	11	5	2	4	8	0	2	0	1	0	0	2	3	5	4	7	3	6	3	5	7	6	8	8	52
JAN.10	5	2	3	2	0	1	0	0	0	0	1	3	3	4	5	6	7	11	7	7	7	5	5	1	113
JAN.11	5	3	1	0	1	2	1	0	0	0	2	2	3	4	1	7	9	6	9	8	4	11	6	11	143
JAN.12	7	3	2	3	0	0	1	0	0	0	0	2	3	6	5	4	5	5	6	8	8	6	8	7	180
JAN.13	7	4	3	3	4	1	1	0	0	0	0	2	4	2	4	7	3	8	8	3	9	7	4	7	107
JAN.14	7	5	2	3	2	1	0	0	0	0	0	1	2	2	6	8	6	5	10	3	5	6	6	3	97
JAN.15	7	4	1	5	0	1	0	0	0	0	0	2	0	3	4	3	7	7	6	5	5	6	7	6	73
JAN.16	0	0	0	0	3	4	3	1	6	3	9	7	4	5	8	12	3	5	3	3	0	2	0	0	99
JAN.17	0	0	0	0	3	5	5	2	4	8	9	5	6	7	8	11	6	3	6	3	1	3	0	0	172
JAN.18	0	0	0	0	3	4	5	4	6	5	4	5	6	9	9	7	10	10	2	1	1	0	1	1	113
JAN.19	0	0	0	0	2	2	6	6	5	7	5	1	4	7	13	10	5	5	1	1	0	0	0	0	90
JAN.20	0	0	0	0	5	2	3	4	5	9	4	5	8	5	8	6	4	4	4	4	2	3	0	0	75
JAN.21	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb
JAN.22	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb	tb
JAN.23	0	0	0	0	2	2	5	5	7	4	6	5	5	4	7	9	7	6	1	6	1	1	0	0	55
JAN.24	0	0	0	0	0	5	5	3	8	1	6	6	7	7	11	5	11	6	4	0	1	0	0	0	121
JAN.25	8	0	0	0	0	4	4	2	6	5	7	9	5	10	8	10	7	6	3	0	3	2	1	0	182
JAN.26	0	0	0	8	2	1	3	5	3	3	4	1	8	4	11	12	6	2	6	0	0	4	0	0	87
JAN.27	0	0	8	8	3	2	4	5	7	5	7	2	8	8	10	8	1	5	5	1	5	4	0	0	86
JAN.28	0	0	0	0	1	5	3	2	7	6	5	9	2	5	7	5	6	6	8	4	3	2	2	0	86
JAN.29	0	0	0	0	2	4	6	3	4	5	4	5	3	12	8	8	8	0	6	2	2	3	0	0	77
JAN.30	0	0	0	1	1	4	6	5	10	5	4	7	8	6	4	11	3	8	6	7	3	1	1	0	189
JAN.31	0	0	0	1	1	2	9	3	2	3	3	4	6	9	3	5	3	6	2	4	3	3	1	0	21
FEB. 1	0	0	1	0	1	5	4	3	3	3	3	6	4	8	4	10	8	9	4	0	1	1	3	2	62
FEB. 2	0	0	0	0	2	3	3	6	5	2	4	7	6	6	8	17	4	7	4	7	3	0	0	0	161
FEB. 3	0	0	0	0	1	5	3	3	3	6	6	7	9	3	13	6	6	6	3	4	1	2	2	3	135
FEB. 4	0	0	5	2	2	5	7	1	3	6	1	1	1	2	0	5	2	6	2	0	0	0	0	0	4
FEB. 5	0	0	0	0	2	4	1	4	5	8	8	6	5	12	6	4	9	4	1	1	1	1	1	0	129
FEB. 6	0	0	0	0	2	2	4	5	4	6	3	9	9	3	7	3	7	0	2	3	7	4	1	0	47
FEB. 7	0	0	0	1	3	5	4	7	2	3	3	8	12	4	9	9	2	2	2	7	2	1	1	2	107
FEB. 8	0	0	0	0	2	2	3	3	7	5	4	6	7	6	6	7	11	3	5	1	4	1	4	0	88
FEB. 9	0	0	0	1	2	5	3	2	6	5	5	3	6	6	7	7	5	7	10	2	1	2	0	0	62
FEB.10	0	0	0	1	2	3	5	1	5	4	3	9	6	7	7	5	13	1	1	7	4	1	1	1	102
FEB.11	0	0	0	3	1	3	7	2	6	9	5	6	4	4	6	11	3	6	3	1	6	2	2	0	82
FEB.12	0	0	0	3	3	3	3	5	5	5	1	8	4	4	11	5	10	6	6	3	3	4	1	0	95
FEB.13	0	0	0	1	4	3	3	3	4	5	3	9	2	9	9	14	8	5	1	4	4	3	3	3	160
FEB.14	0	0	0	1	3	2	4	5	4	4	5	9	6	6	6	13	8	2	12	2	3	1	3	4	166
FEB.15	0	1	2	0	3	4	2	4	5	5	4	6	12	7	9	4	5	11	8	7	1	0	0	0	180
FEB.16	0	0	0	0	2	1	2	2	5	4	4	11	2	14	8	4	11	8	6	3	2	0	0	0	185
FEB.17	0	0	0	2	1	4	4	6	2	7	5	5	4	9	10	11	7	3	5	4	1	4	1	0	132
FEB.10	0	0	0	0	2	3	3	3	5	3	3	2	6	5	16	5	6	9	3	10	7	1	2	0	156
FEB.19	0	0	0	1	3	3	4	5	4	1	6	3	5	4	9	2	8	6	8	1	6	1	2	4	38
FEB.20	0	0	0	0	2	2	3	3	4	5	8	2	5	5	4	15	10	6	4	2	2	1	1	0	86

TOTAL IN LINE WAITING TIME = 4917 HOURS

A 95 percent confidence interval of cumulative in line waiting time in hours per day is (7.93,13.29) for Dong Jing Chen Forestry Bureau with a standard error of 9.53 hours. For Mu Ling Forestry Bureau, the confidence interval is (86.71,113.98) with a standard error of 47.47 hours. Log trucks cumulative waiting time per day at Mu Ling Forestry Bureau indicates that the total waiting time is equivalent to 3.6 trucks wasted every day (or every twenty-four hours).

Admittedly, log truck waiting in line for unloading at the timber yard is not only a problem of the transportation company. Because of inefficiency in communication, the log trucks waiting in line for loading at landings also happens. Sometimes the loading machine breaks down and the log trucks have to go back without loads.

According to the report of Central Forestry Bureau of Hei Long Jiang province, the annual cost due to waiting in line for loading and unloading and log trucks returning without loads, is 5 million to 6 million Chinese Yuns, about 1.35 to 1.62 million U.S dollars.

The investigation of the Dong Jing Chen and Mu Ling Forestry Bureaus, Hei Long Jiang province, suggest that trucks waiting in line for loading at landings and coming back from landings without loads are minor problems. In some other forestry bureaus, however, log trucks waiting in line for loading is not neligible.

The timber yard can be considered the center activities from a shedule point of view. The timber yard is connected with all landings and there are many possible points of departure time to landings and there are many possible points of arrival time from landings. Therefore, the timber yard is a logical point at which to make a schedule.

2. OBJECTIVES

(1) Solve the scheduling problem subject to the assumptions which are appropriate to timber yards in China.

(2) Develop a technique to generate the network using computer.

(3) Demonstrate the use of the method with small example problems and with real world situations.

(4) Discuss statistical techniques to determine upper and lower bounds for production rates and travel time.

3. LITERATURE REVIEW

Bellman et al. (1982) classify scheduling problems into three groups, (a) sequencing or job scheduling problems, (b) project scheduling problems, and (c) assembly-line balancing problems. The objective of the job scheduling problem is to determine the sequence of jobs which a machine or machines should follow to minimize processing time. By classifying the sequencing problem according to the number of machines, the ordering problem can be categrized as fiure 1.

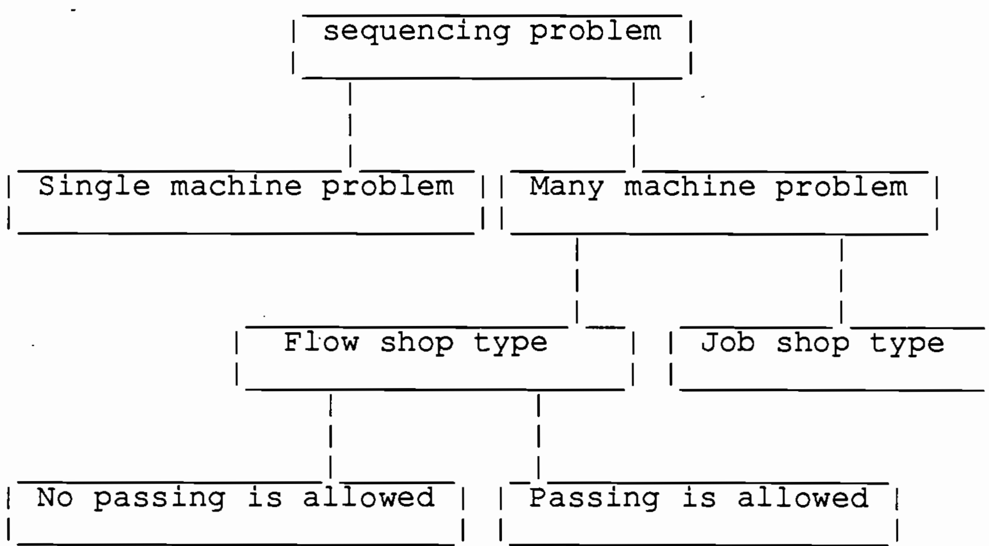

Figure 1. Classification of sequencing problem.

The project scheduling problem is concerned with the coordination of activities to minimize completion times with or without resource constraints. Examples include CPM and PERT. The objective of the assembly line balancing problem is to decide the minimum number of work stations required to minimize total production time. Dynamic programming, branch and bound methods and heuristic methods have been used.

The truck scheduling problem in this paper is somewhat similar to the "fixed schedule minimal fleet problem" considered by Levin (1969) in routing and scheduling problems for airline transportation. According to Levin (1969) the problem first appeared in the Operations Research literature in 1954 as a Transportation problem. Levin proposed both mixed integer programming and variations of maximal flow network approaches to the problem.

Wren (1981) in reviewing the use of computers to schedule buses classifies the schedule as either route based or network based. The objective of the route approach is to link arrivals and departures efficiently at terminals. The routes are generally fixed and repeat themselves with a high frequency. Network based methods, on the other hand, schedule trips over a wide variety of routes.

Network flow algorithms have been popular for solving optimization problems of particular structure due to their superior speed for special applications. Kennington and Helgason (1980) divide network flow models into two types (1) specializations of the primal simplex method and (2) primal-dual methods. They catagorize the out-of-kilter algorithm developed by Delbert Fulkerson as a primal-dual method. There has been considerable discussion about the question of under what conditions the primal simplex method is faster than the primal-dual approach (Barr et al. 1974 and Hatch 1975). Smith (1982) presents a good introduction into the theory of the out-of-kilter algorithm and includes a computer code in BASIC and PASCAL. The out-of-kilter algorithm is discussed in more detail in the Appendix.

Among the network based algorithms, Wren notes that at the Workshop on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transport (Chicago, 1975) Bennington and Rebibo presented a paper using the out-of-kilter algorithm by Fulkerson and Ford. Unfortunately, no proceedings for this workshop were published. The problem addressed by Bennington and Rebibo was to find the cheapest set of paths from the bus garage in the morning to the garage in the evening and covering all nodes in the network. The program handled one type of vehicle and garage. Subsequent work extended the method to several garages. The Bennington and Rebibo problem differs from the log truck scheduling problem in this project in that we must allocate the trucks to landings in such a way as to meet a daily production target.

Although many of the transportation models have been deterministic, Reese (1974) used the out-of-kilter algorithm for investigating the effects of uncertainty in supply and demand nodes for a forest products distribution problem. In the chance constrained problem investigated by Reece, the supply and demand variables were independently and normally distributed. Bounds on the supply and demand nodes were established by requiring flows which resulted in a required probability level.

5. DISCUSSION OF ASSUMPTIONS

In this paper the following assumptions are made:

(1) Under the same road and weather conditions, the speed and variable cost of log trucks are the same.

(2) The traveling time between the landing and timber yard is known. It includes the time from the timber yard to the landing, the time waiting for loading, and the time from the landing to the timber yard.

(3) The amount of tree length available and left in the previous day at each landing is known so a truck allocation can be made for the next day.

(4) Yesterday's schedule is known or at least the arrival times of the trucks currently out is known.

(5) The day is divided into 24 one-hour segments. The smaller the time interval the more precise the estimates, but the less confident we are of truck arrival times during the time segment. To introduce the problem I have arbitrarily used one hour segments. I will discuss more about time interval selection later.

(6) A truck waiting at the timber yard with load has cost which is proportional to the time spent.

(7) Because a truck parked at the parking lot can be used to do something else before it is assigned to another trip to a landing, we assume that no cost takes place if the truck is parked at the parking lot.

4. POSSIBLE SOLUTION METHODS

Several ways to solve this problem have been proposed. Researchers in the Department of Logging Engineering, Northeastern Forestry University, P. R. C., have suggested replacing the old unloading machines by large overhead grapple cranes which can be used to store tree lengths somewhere nearby. This proposal is feasible only in those timber yards with space to store timber. Most of the timber yards, however, do not have available space. In addition, this will incur extra cost and the disruption costs during rebuilding of timber yards would be very expensive. Few forestry bureaus will accept this due to delivery schedules.

Of course, maximum flow of the operation line of the timber yard could be increased, but unfortunately, almost the same requirements are needed as replacing unloading machines.

Another way to solve this problem which can avoid expensive timber yard investments is to control flow to timber yard by reasonably arranging arrivals of log trucks at the timber yard to try to make the distribution of log truck arrivals over time evenly. In this way, the log trucks will be unloaded without waiting.

6. MODEL FORMULATION

6.1 Network Model

It is assumed that the readers are familar with network diagrams and nomenclature. The first node in the model represents the location of the parking lot or garage of the transportation company. This node is connected with the nodes which represent landings. Depending on how many working shifts are worked at each landing, every landing node either is connected with twenty four or sixteen departure nodes. These departure nodes represent log truck departure times at the parking lot of the transportation company. Depending on the number of shifts at the timber yard, there are also either twenty-four or sixteen arrival time nodes. Unlike the other nodes, departure and arrival nodes represent specific time intervals. Therefore, each of those nodes corresponds to the center of a specific time interval. For example, if a node represents time interval (12:30, 13:30), then, the node corresponds to 13 o'clock. The departure nodes and arrival nodes are connected to one another according to the traveling time from each landing to timber yard. If traveling time for landing **i** is **m** hours and the departure node from this landing is **n** o'clock, then, this departure node will be connected to the arrival node (n+m) o'clock. All arrival nodes are connected with the node which represents the timber yard. This timber yard node is connected with a dummy node and the dummy node is connected with the first node, parking lot node (figure 4 on page 19).

connected with a dummy node and the dummy node is connected with the first node, parking lot node (figure 4 on page 19).

Arcs which connect the parking lot node and landing nodes have two bounds each, lower and upper. These bounds are derived by considering the minimum and maximum amounts of wood available in truck loads per day at each landing. There is another value associated with each arc; the total cost per trip associated with each particular landing. For the arcs which connect each landing and departure time nodes, we set the lower bound to zero, and the upper bound equal to the rate of loading of the loading machines at each landing (number of trucks can be loaded per hour). For arcs linking departure time nodes and arrival time nodes, the lower bounds are set to zeroes and the upper bounds are set to the large numbers. The costs are set to zeroes also.

The lower and upper bounds of arcs between arrival time nodes and timber yard node are set to zero and the maximum flow of the assembling line of the timber yard, respectively. We want the number of truck arrivals at the timber yard in each time interval to be constant to promote efficiency of transhipment to railroad (Figure 2).

Figure 2. Desirable distribution of truck arrivals for 24 hour per day operation with equal crew size on each shift.

According to a particular desired distribution of log truck arrivals at the timber yard, the upper bounds of these arcs can be set to corresponding values to meet the reqirements. For example, we may hope the distribution looks as shown in Figure 2.

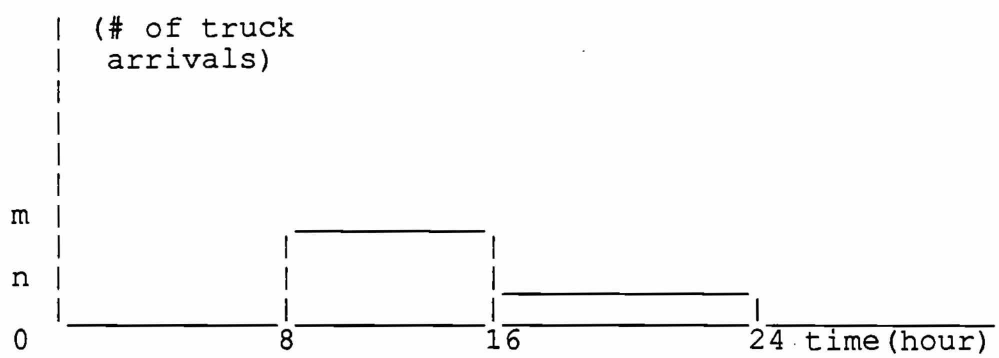

Figure 3. Desirable distribution of truck arrivals with a two shits operation when the early shift has a larger crew operation.

If the sum of the lower bounds of arcs from the parking lot to landings is greater than the sum of the upper bounds of the arcs from the arrival time nodes to the timber yard, the network has no feasible solution. In order to avoid this, we provide a high cost return route without limits between the landings and parking lot.

At the expense of a little bit of inefficiency, a model which is suitable for any kind of crew schedule (three shifts, two shifts) can be set up. In such a model, the number of all the departure nodes associated with each landing are twenty-four and so are the number of arrival nodes. If the real work hours are sixteen instead of twenty -four for landings, for example, set upper bounds of the corresponding arcs equal to zero.

Since it is not feasible to draw a complete network diagram on a single page for a practical problem in China due to the number of arcs and nodes, I will demonstrate the technique on a smaller problem. Let's assume that there are only five landings, working time is five hours at each landing and seven hours at the timber yard. The input related with landings is as follows:

Table 3. INPUT OF A SAMPLE NETWORK.

landing identifict^n number	lower bounds (trips)	upper bounds (trips)	cost ($/ trip)	traveling time (hours)	loading rate (trucks/hr)
1	6	8	15	4	3
2	10	12	18	5	2
3	8	10	20	6	2
4	6	8	25	6	1
5	5	10	20	7	2

The revenue per truck load is $50 and unloading rate is 5 trucks per hour. The diagram of the network will be as follows:

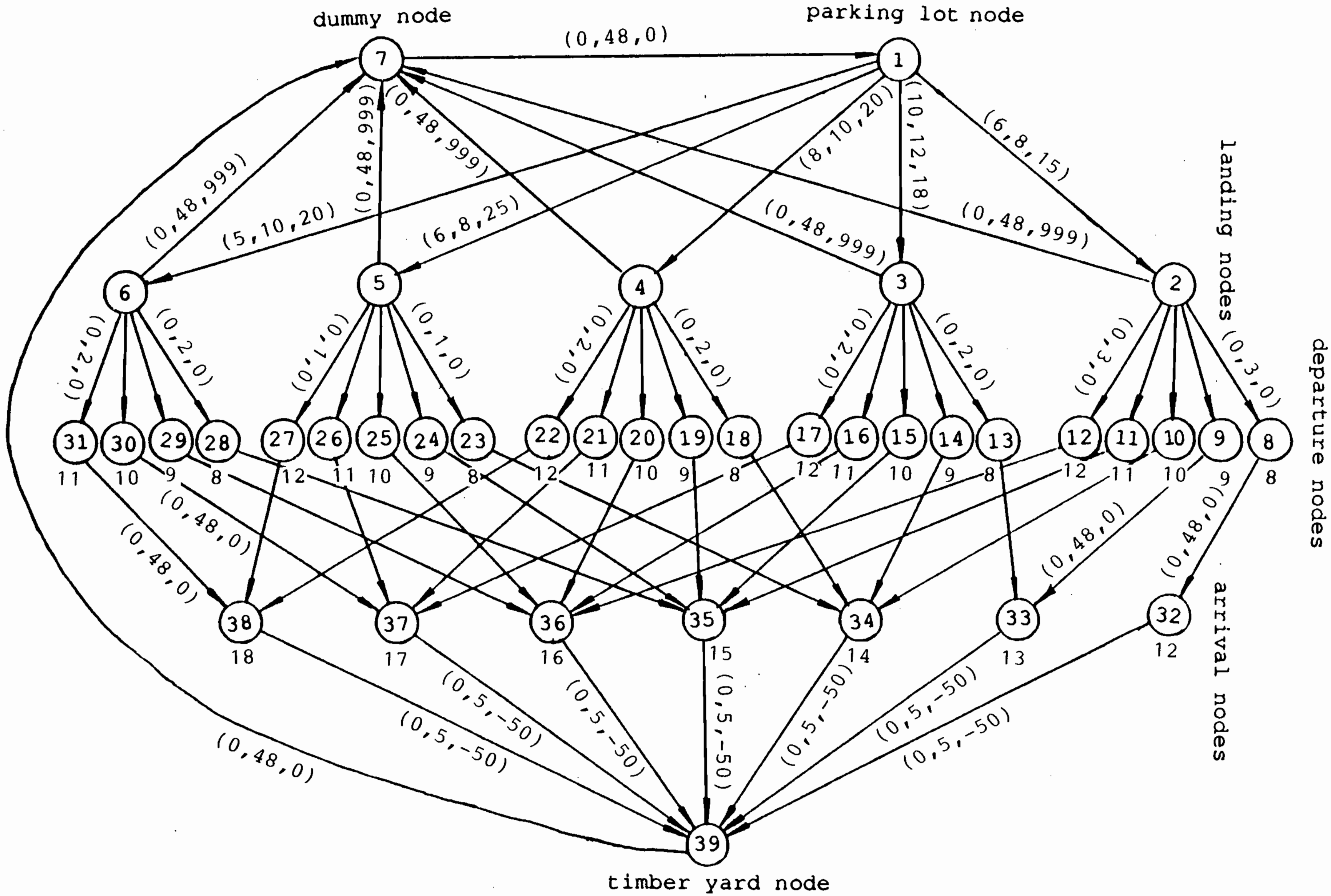

Figure 4. A sample network.

6.2 Mathematical Model

$$\text{minimize} \quad z=\Sigma[c(i,j)*x(i,j)]$$

$$\text{subject to:} \quad -\sum_i x(i,j)+\sum_k x(j,k)=0$$

$$0 \le l(i,j) \le x(i,j) \le u(i,j)$$

where x(i,j) is flow from node i to j;

c(i,j) is the cost per unit of flow in arc(i,j);

l(i,j) and u(i,j) are lower and upper bounds of arc(i,j)

For previous example, the mathematical model is as follows:

$$\text{MINIMIZE } z=15*X_{1,2}+18*X_{1,3}+20*X_{1,4}+25*X_{1,5}+20*X_{1,6}$$
$$+999*X_{2,7}+999*X_{3,7}+999*X_{4,7}+999*X_{5,7}+999*X_{6,7}$$
$$-50*X_{32,39}-50*X_{33,39}-50*X_{34,39}-50*X_{35,39}$$
$$-50*X_{36,39}-50*X_{37,39}-50*X_{38,39}$$

$$\text{s.t.: } X_{1,2}+X_{1,3}+X_{1,4}+X_{1,5}+X_{1,6}-X_{7,1}=0$$
$$X_{2,8}+X_{2,9}+X_{2,10}+X_{2,11}+X_{2,12}+X_{2,7}-X_{1,2}=0$$
$$X_{3,13}+X_{3,14}+X_{3,15}+X_{3,16}+X_{3,17}+X_{3,7}-X_{1,3}=0$$
$$X_{4,18}+X_{4,19}+X_{4,20}+X_{4,21}+X_{4,22}+X_{4,7}-X_{1,4}=0$$
$$X_{5,23}+X_{5,24}+X_{5,25}+X_{5,26}+X_{5,27}+X_{5,7}-X_{1,5}=0$$
$$X_{6,28}+X_{6,29}+X_{6,30}+X_{6,31}+X_{6,7}-X_{1,6}=0$$
$$X_{8,32}-X_{2,8}=0$$
$$X_{9,33}-X_{2,9}=0$$

$$x_{10,34}-x_{2,10}=0$$
$$x_{11,35}-x_{2,11}=0$$
$$x_{12,36}-x_{2,12}=0$$
$$x_{13,33}-x_{3,13}=0$$
$$x_{14,34}-x_{3,14}=0$$
$$x_{15,35}-x_{3,15}=0$$
$$x_{16,36}-x_{3,16}=0$$
$$x_{17,37}-x_{3,17}=0$$
$$x_{18,34}-x_{4,18}=0$$
$$x_{19,35}-x_{4,19}=0$$
$$x_{20,36}-x_{4,20}=0$$
$$x_{21,37}-x_{4,21}=0$$
$$x_{22,38}-x_{4,22}=0$$
$$x_{23,34}-x_{5,23}=0$$
$$x_{24,35}-x_{5,24}=0$$
$$x_{25,36}-x_{5,25}=0$$
$$x_{26,37}-x_{5,26}=0$$
$$x_{27,38}-x_{5,27}=0$$
$$x_{28,35}-x_{6,28}=0$$
$$x_{29,36}-x_{6,29}=0$$
$$x_{30,37}-x_{6,30}=0$$
$$x_{31,38}-x_{6,31}=0$$
$$x_{32,39}-x_{8,32}=0$$
$$x_{33,39}-x_{9,33}-x_{13,33}=0$$
$$x_{34,39}-x_{10,34}-x_{14,34}-x_{18,34}-x_{23,34}=0$$
$$x_{35,39}-x_{11,35}-x_{15,35}-x_{19,35}-x_{24,35}-x_{28,35}=0$$
$$x_{36,39}-x_{12,36}-x_{16,36}-x_{20,36}-x_{25,36}-x_{29,36}=0$$

$x_{37,39}-x_{17,37}-x_{21,37}-x_{26,37}-x_{30,37}=0$

$x_{38,39}-x_{22,38}-x_{27,38}-x_{31,38}=0$

$x_{39,7}-x_{32,39}-x_{33,39}-x_{34,39}-x_{35,39}-x_{36,39}$

$-x_{37,39}-x_{38,39}=0$

$x_{7,1}-x_{2,7}-x_{3,7}-x_{4,7}-x_{5,7}-x_{6,7}-x_{39,7}=0$

$6 \leq x_{1,2} \leq 8$

$10 \leq x_{1,3} \leq 12$

$8 \leq x_{1,4} \leq 10$

$6 \leq x_{1,5} \leq 8$

$5 \leq x_{1,6} \leq 10$

$0 \leq x_{2,7} \leq 48$

$0 \leq x_{3,7} \leq 48$

$0 \leq x_{4,7} \leq 48$

$0 \leq x_{5,7} \leq 48$

$0 \leq x_{6,7} \leq 48$

$0 \leq x_{2,8} \leq 3$

$0 \leq x_{2,9} \leq 3$

$0 \leq x_{2,20} \leq 3$

$0 \leq x_{2,11} \leq 3$

$0 \leq x_{2,12} \leq 3$

$0 \leq x_{3,13} \leq 2$

$0 \leq x_{3,14} \leq 2$

$0 \leq x_{3,15} \leq 2$

$0 \leq x_{3,16} \leq 2$

$0 \leq x_{3,17} \leq 2$

$0 \leq x_{4,18} \leq 2$

$0 \leq x_{4,19} \leq 2$

$$0 \leq X_{4,20} \leq 2$$

$$0 \leq X_{4,21} \leq 2$$

$$0 \leq X_{4,22} \leq 2$$

$$0 \leq X_{5,23} \leq 1$$

$$0 \leq X_{5,24} \leq 1$$

$$0 \leq X_{5,25} \leq 1$$

$$0 \leq X_{5,26} \leq 1$$

$$0 \leq X_{5,27} \leq 1$$

$$0 \leq X_{6,28} \leq 2$$

$$0 \leq X_{6,29} \leq 2$$

$$0 \leq X_{6,30} \leq 2$$

$$0 \leq X_{6,31} \leq 2$$

$$0 \leq X_{8,32} \leq 48$$

$$0 \leq X_{9,33} \leq 48$$

$$0 \leq X_{10,34} \leq 48$$

$$0 \leq X_{11,35} \leq 48$$

$$0 \leq X_{12,36} \leq 48$$

$$0 \leq X_{13,33} \leq 48$$

$$0 \leq X_{14,34} \leq 48$$

$$0 \leq X_{15,35} \leq 48$$

$$0 \leq X_{16,36} \leq 48$$

$$0 \leq X_{17,37} \leq 48$$

$$0 \leq X_{18,34} \leq 48$$

$$0 \leq X_{19,35} \leq 48$$

$$0 \leq X_{20,36} \leq 48$$

$$0 \leq X_{21,37} \leq 48$$

$$0 \leq X_{22,38} \leq 48$$

$0 \leq X_{23,34} \leq 48$

$0 \leq X_{24,35} \leq 48$

$0 \leq X_{25,36} \leq 48$

$0 \leq X_{26,37} \leq 48$

$0 \leq X_{27,38} \leq 48$

$0 \leq X_{28,35} \leq 48$

$0 \leq X_{29,36} \leq 48$

$0 \leq X_{30,37} \leq 48$

$0 \leq X_{31,38} \leq 48$

$0 \leq X_{32,39} \leq 48$

$0 \leq X_{33,39} \leq 48$

$0 \leq X_{34,39} \leq 48$

$0 \leq X_{35,39} \leq 48$

$0 \leq X_{36,39} \leq 48$

$0 \leq X_{37,39} \leq 48$

$0 \leq X_{38,39} \leq 48$

$0 \leq X_{39,7} \leq 48$

$0 \leq X_{7,1} \leq 48$

7. SOLUTION PROCEDURE

7.1 Input of the Models and Definitions

7.1.1 Definitions

(1) Mean unloading rate is the number of loads of log trucks which can pass through the assembly line of the timber yard per hour.

(2) The mean unloading time equals to 1/(mean unloading rate).

(3) Mean traveling time of landing i is the sum of three time elements: mean traveling time from the timber yard to landing i, mean loading time of landing i and mean traveling time from that landing back to timber yard.

(4) Cost associated with each landing may include labor cost, fuel consumption, truck depreciation and so on.

7.1.2 Input For the Models

The input data needed here are as follows: (1) the minimum and the maximum number of trips of log trucks needed by every landing in a day, denoted by l(i) and u(i) respectively, (2) mean round trip traveling time associated with each landing, denoted by t(i); (3) total cost per trip for landing i, denoted by c(i); (4) mean loading rate associated with landing i, denoted by load rate(i); (5) minimum and maximum number of trucks that can arrive at the timber yard for time interval k, denote by lb(k) and ub(k).

7.2 Generation of Network

7.2.1 Necessity of Network Generation

It is important to generate the networks mechanically with the computer. This makes the technique usable for people who do not know network formulations but wish to use this method. Even for the people who do know network theory , it is difficult to type in long network files and avoid mistakes. Since many things are subject to change, the final results, i.e. a time table must be able to be updated easily. Therefore, unless the network is to be generated by computer, any effort which relies on typing a network file into computer will be too slow for practical applications.

7.2.2 Network generation

Because of only one timber yard in the system, it is very easy to generate the network mechanically.

Let **lndg** denote the number of landings which are in the operation, **n** denote the number of nodes in network and **m** the number of arcs in the network. Base on twenty-four possible schedule time, we have:

N=25*lndg+27

M=50*lndg+26

Now, all nodes and their meanings can be defined as follows:

node(1) ....parking lot

node(2) through node(1+lndg). ....landing nodes

node(2+lndg).................... ......dummy node

node(3+lndg) through node(3+lndg+24*lndg-1)

or node(3+lndg) through node(25*lndg+2)

....departure nodes

node(25*lndg+30 through node(25*lndg+3+23)

or node(25*lndg+3) through node(25*lndg+26)

....arrival nodes

node(25*lndg+27).....................timber yard node

Unlike other nodes, the time nodes (departure nodes and arrival nodes), node(lndg+3) through node(25*lndg+26), need to be assigned a value each which is the time, for example, 10 o'clock, 2 o'clock and so on.

Now we are ready to connect the nodes into a network by defining arcs.

(1) Arcs which connect the parking lot and landing nodes: there are totally (lndg) arcs. For the i-th arc, its start node, end node, lower bound, upper bound and cost are I(i)=i, J(i)=i+1, L(i), U(i), and C(i) respectively, where L(i), U(i) and C(i) are associated with the i-th landing and i=1,2,...,lndg.

(2) Arcs linking dummy node and landing nodes: there are a total of (lndg) arcs labeled as arc (lndg+1) through arc (2*lndg). The start node, end node, lower bound, upper bound and cost are

I(i)=i-lndg+1

J(i)=2+lndg

L(i)=0

U(i)=truck

C(i)=999

respectively.

Where **truck** is the number of maximum total trips needed to finish all landing transportation tasks for the day in question

i=(lndg+1),...,2*lndg.

(3) Arcs which connect landing nodes and departure nodes: there are (24*lndg) arcs. These arcs are labeled arc(2*lndg+1) through arc(26*lndg+1). For k1-th arc of these arcs, the start nodes, end nodes, lower bounds, upper bounds and cost are

I(k1)=i+1	where i=1,..., lndg
J(k1)=2+lndg+k2	k1=2*lndg,...,2*lndg+24*lndg
L(k1)=0	k2=1,...,24*lndg
U(k1)=load rate(i)	
C(k1)=0	

(4) Arcs which connect departure and arrival nodes: there are (24*lndg) arcs in this set of arcs. They are arc (26*lndg+1) through arc (26*lndg+1+24*lndg). The k3-th arc of this set of arcs can be defined as follows:

I(k3)=lndg+2+j+24*(i-1)

J(k3)=25*lndg+3+[T(i)+(j-1)] mod 24

L(k3)=0

U(k3)=truck

C(k3)=0

where k3=26*lndg+1,...,24*lndg+26*lndg+1

i=1,...,lndg

j=1,...,24

(5) Arcs which connect arrival and timber yard nodes: there are twenty four arcs in this set. They are arc (50*lndg+1) through arc (50*lndg+24). The k4-th arc are defined as follows:

I(k4)=25*lndg+2+i

J(k4)=25*lndg+27

L(k4)=lb(i)

U(k4)=ub(i)

C(k4)=-revenue

where k4=50*lndg+1,...,50*lndg+24

i=1,...,24

lb(i) and ub(i) are lower and upper bounds of confidence interval of unloading rate (truck per hour) of the i-th time interval.

(6) Arc linking timber yard node and dummy node: this arc is defined as:

I(50*lndg+25)=25*lndg+27

J(50*lndg+25)=lndg+2

L(50*lndg+25)=0

U(50*lndg+25)=truck

c(50*lndg+25)=0

(7) Arc linking dummy node and parking lot node:

I(50*lndg+26)=lndg+2

J(50*lndg+26)=1

```
L(50*lndg+26)=0
U(50*lndg+26)=truck
C(50*lndg+26)=0
```

7.3 Optimal Solution of the Network

The "out-of-kilter" algorithm was used to find the network solution. The use of the computer program by SMITH (1982) was originally attempted. This program, however, was found to not work. Consequently, some modifications had to be made. The modifications to Smith's computer code and a brief review of the "out-of kilter" algorithm, are included in Appendix A.

7.4 Interpretation of the network solution

The form of the solution of the network is as follows:

Table 4. FORMAT OF THE NETWORK SOLUTION.

Time of .departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
I1(i)	J1(i)	L1(i)	U1(i)	X1(i,j)
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
In(i)	Jn(i)	Ln(i)	Un(i)	Xn(i,j)

Of course, the solution like this can not be used directly. It is translated in a report writer, because the networks become large and complex. Like network generation, it is convenient to interpret the network results with the computer. The results can be translated into a time table from which one can read the following: when, to which landing, and how many trucks should be sent out. The time table is in following form:

Table 5. FORMAT OF THE TIME TABLE.

Time of .departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
D1	N1	L1	B1	M1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
Dk	Nk	Lk	Bk	Mk

7.5 Truck Inventory Analysis

7.5.1 Dynamic truck inventory analysis

The column under the title of "number of trucks" in the time table produced in section 7.4 are actually the trips. This is because each truck can complete more than one trip depending on the traveling time or the traveling distance.

Of course, it is important to know how many trucks are needed to conduct the transportation plan developed above.

Since we know the existing log truck inventory, the number of log trucks which will be sent out at time interval i, the number of trucks that will be back to the timber yard at the same time interval, we can calculate the log truck inventory in that time interval. The number of log trucks needed is equal to the sum of absolute value of negative inventory of the i-th time interval.

$$INV(i-1)=INV_0$$

$$INV(i)=INV(i-1)+ARR(i)-DEP(i)$$

$$\text{TRUCKS NEEDED}=-\Sigma[INV(i)] \quad \text{if } INV(i)<0$$

Where $i=1,\ldots,24$

INV_0 is the initial inventory

INV(i) is the log truck inventory at the time interval i

ARR(i) is the number of log trucks which arrive at the timber yard at time interval i

DEP(i) is the number of log trucks which depart in time interval i

In this example, the traveling time associated with each landing is assumed to be less than 24 hours in the truck inventory anlysis section. This is because it is easier for the computer to distinguish truck arrivals at the same time intervals but different days.

7.5.2 Determination of the number of trucks needed

In the dynamic truck inventory analysis process, whenever the truck inventory becomes negative, for example, IVN(i)=-3, three more trucks have to be sent out from the parking lot. Therefore, the absolute value of the sum of INV(i) will be the total trucks which we need.

8. RESULTS

8.1 Example 1

The input of the model, as pointed out in the previous section, includes the information for the landings and the timber yard. In this example, we assume that the timber yard will pay $50 to the transportation company if the timber yard receives one delivered truck load. The mimimum unloading rate of the unloading machines of the timber yard is 4 trucks per hour. This means that the same number of unloading machines are in operation through out the day. The desirable truck arrival distribution is shown in figure 5. There are no trucks in transit at the beginning of the day.

Figure 5. Desirable truck arrival distribution.

The landing identification number (column 1), trips which must be completed (lower bound in column 2), the maximum number of trips that can be completed each day (upper bound in column 3), the cost per trip (column 4), the traveling time and the loading rate for each landing (column 5 and 6) are given as follows:

Table 6. INPUT OF EXAMPLE 1.

landing identifict^n number	lower bounds (trips)	upper bounds (trips)	cost ($/ trip)	traveling time (hours)	loading rate (trucks/hr)
1	16	20	34	6	2
2	10	15	20	4	2
3	25	30	40	10	4

The output from the model includes the solution of the network, the recommended log truck schedule, log truck inventory analysis table and the number of log trucks needed to carry out the recommended log truck schedule. The output for example 1 is shown in table 7.

Table 7. THE OUTPUT OF THE NETWORK OF EXAMPLE 1.

Time of .departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
2	2	1	8	2
3	2	1	9	2
4	2	1	10	2
5	2	1	11	2
6	2	1	12	2
7	2	1	13	2
8	2	1	14	2
9	2	1	15	2
14	1	1	20	1

Table 7. THE OUTPUT OF THE NETWORK OF EXAMPLE 1. (CONTINUED)

Time of departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
15	1	1	21	1
16	2	1	22	2
4	2	2	8	2
5	2	2	9	2
6	2	2	10	2
7	2	2	11	2
8	2	2	12	2
9	2	2	13	2
17	1	2	21	1
18	2	2	22	2
4	2	3	14	2
5	2	3	15	2
6	4	3	16	4
7	4	3	17	4
8	4	3	18	4
9	4	3	19	4
10	3	3	20	3
11	2	3	21	2
13	4	3	23	4

The network output in table 7 is part of the direct network output. The part for which the flow is zero is exluded here.

Table 8. RECOMMENDED LOG TRUCK SCHEDULE OF EXAMPLE 1.

Time of .departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
2	2	1	8	2
3	2	1	8	2
4	2	3	9	2
4	2	2	9	2
4	2	1	10	2
5	2	3	10	2
5	2	2	11	2
5	2	1	11	2
6	4	3	12	2
6	2	2	12	2
6	2	1	13	2
7	4	3	13	2
7	2	2	14	2
7	2	1	14	2
8	4	3	15	2
8	2	2	15	2
8	2	1	16	4
9	4	3	17	4
9	2	2	18	4
9	2	1	19	4

Table 8. THE RECOMMENDED LOG TRUCK SCHEDULE OF EXAMPLE 1. (CONTINUED)

Time of departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
10	3	3	20	3
11	2	3	20	1
13	4	3	21	2
14	1	1	21	1
15	1	1	21	1
16	2	1	22	2
17	1	2	22	2
18	2	2	23	4

The "Recommended log truck schedule " is the time table which can be used directly by the log truck manager. This schedule is developed by sorting the departure time and arrival time of table 7 by hour. At 2 o'clock, as one can see, 2 log trucks should be sent to landing 1. At 3 o'clock, another 2 log trucks will be sent to landing 1. At 4 o'clock, 6 log trucks will be sent out, 2 to landing 1, 2 to landing 2 and 2 to landing 3.

Table 9 gives the log truck manager an idea about how many log trucks are on hand, how many log truck will be sent out and how many will come back at a given time interval. The fourth column is calculated by subtracting the third column from the second column.

Table 9. TRUCK INVENTORY ANALYSIS.

Time (o'clock)	# of trucks departed	# of truck arrivals	Difference between trucks dep. & arr.
0	0	0	0
1	0	0	0
2	2	0	2
3	2	0	2
4	6	0	6
5	6	0	6
6	8	0	8
7	8	0	8
8	8	4	4
9	8	4	4
10	3	4	-1
11	2	4	-2
12	0	4	-4
13	4	4	0
14	1	4	-3
15	1	4	-3
16	2	4	-2
17	1	4	-3
18	2	4	-2
19	0	4	-4
20	0	4	-4
21	0	4	-4
22	0	4	-4
23	0	4	-4

Total trucks needed = 40

The first two lines of table 9 tell us that at first two time interval, i.e., time interval 0 (which is from 23:30 of yesterday to 0:30) and time interval 1, no trucks should be sent out and no trucks will arrive at the timber yard. At the time interval 2, two log trucks should be sent out, while no log trucks will come back. Therefore, a minimum inventory of two log trucks are needed. In the tine interval 10, on the other hand, three log trucks will be sent out and four log trucks will come back. Therefore, the total log truck inventory is increased by one up to this point. As we can see at the bottom of the table, the total log trucks needed to carry out the recommended schedule is 40 trucks in this example.

8.2 Example 2

Input which is related with the timber yard: (1) The revenue per delivered truck load is $65; (2) A constant unloading rate is 5 vehicles per hour. This means that the lower bounds are zero and the upper bounds are 5 for the arcs representing truck arrivals. The input data related with the landings is in Table 10 and the network output is in Table 11.

Table 10. INPUT OF EXAMPLE 2.

landing identifict^{n} number	lower bounds (trips)	upper bounds (trips)	cost ($/ trip)	traveling time (hours)	loading rate (trucks/hr)
1	12	15	39	12	2
2	20	22	18	7	4
3	15	27	25	9	3
4	30	40	20	6	3
5	8	10	24	5	1

Table 11. THE OUTPUT OF THE NETWORK OF EXAMPLE 2.

Time of .departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
12	2	1	0	2
13	2	1	1	2
14	2	1	2	2
15	2	1	3	2
16	2	1	4	2
17	2	1	5	2
6	1	2	13	1
7	1	2	14	1
8	1	2	15	1
9	1	2	16	1
17	3	2	0	3
18	3	2	1	3

Table 11. THE OUTPUT OF THE NETWORK OF EXAMPLE 2.
(CONTINUED)

Time of departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
19	3	2	2	3
20	3	2	3	3
21	3	2	3	3
22	3	2	5	3
0	3	3	9	3
1	3	3	10	3
2	3	3	11	3
3	2	3	12	2
8	1	3	17	1
9	1	3	18	1
10	1	3	19	1
11	1	3	20	1
12	2	3	21	2
13	3	3	22	3
14	3	3	23	3
23	3	3	8	3
2	2	4	8	2
3	1	4	9	1
4	2	4	10	2
5	2	4	11	2
6	3	4	12	3
7	3	4	13	3
8	3	4	14	3

Table 11. THE OUTPUT OF THE NETWORK OF EXAMPLE 2.
(CONTINUED)

Time of departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
9	3	4	15	3
10	3	4	16	3
11	3	4	17	3
12	3	4	18	3
13	3	4	19	3
14	3	4	20	3
15	2	4	21	2
16	2	4	22	2
17	2	4	23	2
4	1	5	9	1
8	1	5	13	1
9	1	5	14	1
10	1	5	15	1
11	1	5	16	1
12	1	5	17	1
13	1	5	18	1
14	1	5	19	1
15	1	5	20	1
16	1	5	21	1

Table 12. RECOMMENDED TRUCK SCHEDULE OF EXAMPLE 2.

Time of departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
0	3	3	0	3
1	3	3	0	2
2	2	4	1	3
2	3	3	1	2
3	1	4	2	3
3	2	3	2	2
4	2	4	3	3
4	1	5	3	2
5	2	4	4	3
6	3	4	4	2
6	1	2	5	3
7	3	4	5	2
7	1	2	8	2
8	3	4	8	3
8	1	3	9	1
8	1	5	9	1
8	1	2	9	3
9	3	4	10	2
9	1	3	10	3
9	1	5	11	2
9	1	2	11	3
10	3	4	12	3
10	1	3	12	2

Table 12. RECOMMENDED TRUCK SCHEDULE OF EXAMPLE 2.
(CONTINUED)

Time of departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
10	1	5	13	1
11	3	4	13	3
11	1	3	13	1
11	1	5	14	1
12	3	4	14	3
12	2	3	14	1
12	1	5	15	1
13	3	4	15	3
13	3	3	15	1
13	1	5	16	1
14	3	4	16	3
14	3	3	16	1
14	1	5	17	1
15	2	4	17	3
15	1	5	17	1
16	2	4	18	1
16	1	5	18	3
17	2	4	18	1
*12	2	1	19	1
*13	2	1	19	3
*14	2	1	19	1
*15	2	1	20	1

Table 12. RECOMMENDED TRUCK SCHEDULE OF EXAMPLE 2.
(CONTINUED)

Time of .departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
*16	2	1	20	3
*17	2	1	20	1
*17	3	2	21	1
*18	3	2	21	2
*19	3	2	21	2
*20	3	2	22	2
*21	3	2	22	3
*22	3	2	23	2
*23	3	3	23	3

* indicates that the departure time, number of departure trucks and the landing destination occur the day before our planning day.

Table 13. TRUCK INVENTORY ANALYSIS OF EXAMPLE 2.

Time (o'clock)	# of trucks departed	# of truck arrivals	Difference between trucks dep. & arr.
0	3	5	-2
1	3	5	-2
2	5	5	0
3	3	5	-2
4	3	5	-2
5	2	5	-3
6	4	0	4

Table 13. TRUCK INVENTORY ANALYSIS OF EXAMPLE 2.
(CONTINUED)

Time (o'clock)	# of trucks departed	# of truck arrivals	Difference between trucks dep. & arr.
7	4	0	4
8	6	5	1
9	6	5	1
10	5	5	0
11	5	5	0
12	6	5	1
13	7	5	2
14	7	5	2
15	3	5	-2
16	3	5	-2
17	2	5	-3
18	0	5	-5
19	0	5	-5
20	0	5	-5
21	0	5	-5
22	0	5	-5
23	0	5	-5

TOTAL TRUCKS NEEDED = 48

8.3 Example 3

The data used in this example was collected at the LANG-SANG FORESTRY BUREAU by Li Hai Wang in 1986. Unfortunately, not all data needed here can be found in Wang's paper. Therefore, we assume that the loading rate at all landings is 2 trucks per hour. The unloading rate is assumed to be 4 trucks per hour at timber yard. The revenue per delivered truck load is $60. The data associated with landings are shown in following table:

Table 14. INPUT OF EXAMPLE 3.

landing identifict^{n} number	lower bounds (trips)	upper bounds (trips)	cost ($/ trip)	traveling time (hours)	loading rate (trucks/hr)
1	3	4	13	2	2
2	4	5	13	2	2
3	2	3	23	3	2
4	2	3	23	3	2
5	5	6	48	4	2
6	5	6	48	4	2
7	1	2	51	5	2
8	1	2	51	5	2
9	2	3	51	2	2
10	4	5	37	5	2

Table 15. OUTPUT OF EXAMPLE 3.

Time of departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
6	2	1	8	2
7	1	1	9	1
12	1	1	14	1
6	2	2	8	2
7	2	2	9	2
13	1	2	15	1
6	1	3	9	1
7	1	3	10	1
12	1	3	15	1
7	2	4	10	2
13	1	4	16	1
6	1	5	10	1
7	2	5	11	2
8	2	5	12	2
12	1	5	16	1
7	2	6	11	2
8	2	6	12	2
9	1	6	13	1
12	1	6	16	1

Table 15. OUTPUT OF EXAMPLE 3.
(CONTINUED)

Time of departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
8	1	7	13	1
12	1	7	17	1
8	1	8	13	1
12	1	8	17	1
11	1	9	13	1
12	1	9	14	1
15	1	9	17	1
9	2	10	14	2
10	2	10	15	2
11	1	10	16	1

Table 16. RECOMANDED TRUCK SCHEDULE OF EXAMPLE 3.

Time of .departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
6	1	5	8	2
6	1	3	8	2
6	2	2	9	1
6	2	1	9	2
7	2	6	9	1
7	2	5	10	1
7	2	4	10	2
7	1	3	10	1
7	2	2	11	2
7	1	1	11	2
8	1	8	12	2
8	1	7	12	2
8	2	6	13	1
8	2	5	13	1
9	2	10	13	1
9	1	6	13	1
10	2	10	14	2
11	1	10	14	1
11	1	9	14	1
12	1	9	15	2
12	1	8	15	1
12	1	7	15	1
12	1	6	16	1

Table 16. RECOMMENDED TRUCK SCHEDULE OF EXAMPLE 3.
(CONTINUED)

Time of departure (o'clock)	Number of dep. trucks	To landing i	Arrival time (o'clock)	Number of arri. trucks
12	1	5	16	1
12	1	3	16	1
12	1	1	16	1
13	1	4	17	1
13	1	2	17	1
15	1	9	17	1

Table 17. TRUCK INVENTORY ANALYSIS OF EXAMPLE 3.

Time (o'clock)	# of trucks departed	# of truck arrivals	Difference between trucks dep. & arr.
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	6	0	6
7	10	0	10
8	6	4	2
9	3	4	-1
10	2	4	-2
11	2	4	-2

Table 17. TRUCK INVENTORY ANALYSIS OF EXAMPLE 3.
(CONTINUED)

Time (o'clock)	# of trucks departed	# of truck arrivals	Difference between trucks dep. & arr.
12	7	4	3
13	2	4	-2
14	0	4	-4
15	1	4	-3
16	0	4	-4
17	0	3	-3
18	0	0	0
19	0	0	0
20	0	0	0
21	0	0	0
22	0	0	0
23	0	0	0

The total trucks needed = 21

9. CONCLUSION

I have presented a network-based method for solving the log truck scheduling problem for minimizing truck utilization time (cost) while assuring constant arrivals or other designed arrival distributions at the timber yard. A code has been written to generate the network and the solve the network with the "out-of-kilter" algorithm.

With the method used in this paper, besides meeting the daily prodution targets and minimum cost objective, waiting time for truck loading and unloading is controlled. A detailed log truck schedule table and total trucks needed to carry out that schedule can be obtained.

The approach used in this paper can be used to rapidly solve the problem providing flexibility to update the current truck status and solve for a new schedule.

With a given level of certainty in technical aspects, such as that of log truck mechanical condition and road condition, a confidence level for the model can be determined and a precision can be reached.

10. SUGGESTIONS FOR FURTHER RESEACH

Only one type of log truck is included in the model. In some forestry bureaus, there are more than one type of log truck. It is, however, very rare that there are three or more types of log trucks. Most of forestry bureaus have two sizes of log trucks. One is small and the other is large in terms of the load that it can transported. Since the network model presented here assumes that the units of flow are homogeneous, we cannot handle two types of traffic directly. The following heuristic approach might be tried. First, run the program with inputs associated with the small truck, find out the total trucks needed, and adjust the bounds of the arcs linking the parking lot and landings until the number of trucks needed are equal to the trucks available or the number of trucks that we plan to use. Then find out the amount of wood left at each landing and reset new upper bounds of the arcs linking arrival nodes and the timber yard. To do so, we can substract the number of arrivals of the log trucks in the time interval in question from the old upper bound. Run the program provided again and find out the number of big trucks needed. As we know from the section of "results", in this process, we get a time table which we need as well.

The question now is why the small trucks need be considered first. This is because (1) the small trucks are sent to the nearest landings and the big trucks are assigned to the farthest landings is considered to be reasonable and (2) the "out-of-kilter" algorithm used here will pick least expensive routes or shortest routes (if the cost per trip is proportional to the associated distance) in its solution. Therefore, if the small trucks are consided first, they will be assigned to the relatively shortest routes.

Up to this point, one may ask if we can find the optimal solution with this procedure ? This is a good question. Further research is needed to answer it. In other words, we need prove that the overall solution found in this way still is optimal. If it is not optimal, or if it is too slow to find the solution, a different model is needed.

If the first suggestion is suitable, the secant method can be used to find the number of small trucks needed and work out the time table at the same time. The flow chart shown in figure 6 and figure 7 may help.

Linear programming is another possible approach. The advantage of it is that it is easy to conside different types of truck. The problem is (1) the matrix generation is larger and more complicated and (2) the larger problem will take longer time to solve.

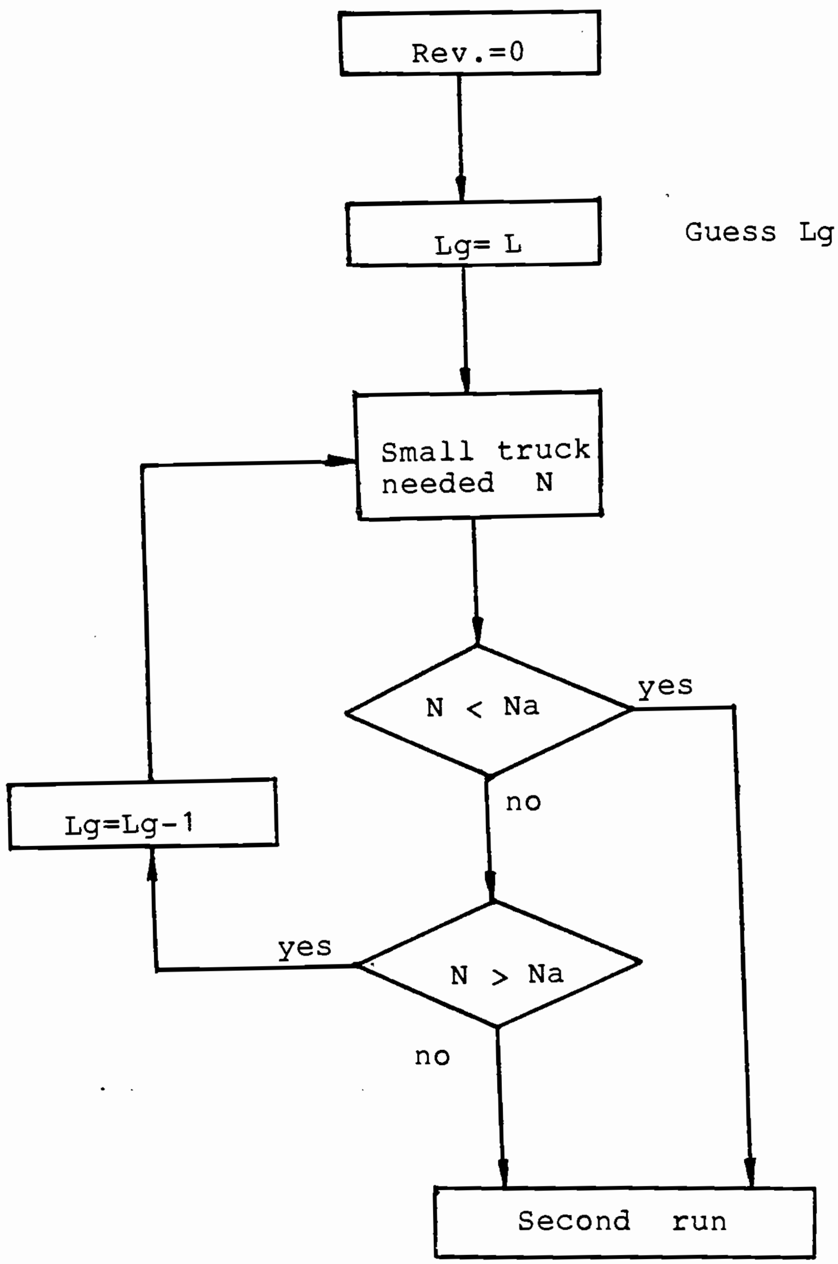

Figure 7. Flow chart for two truck types in category 1.

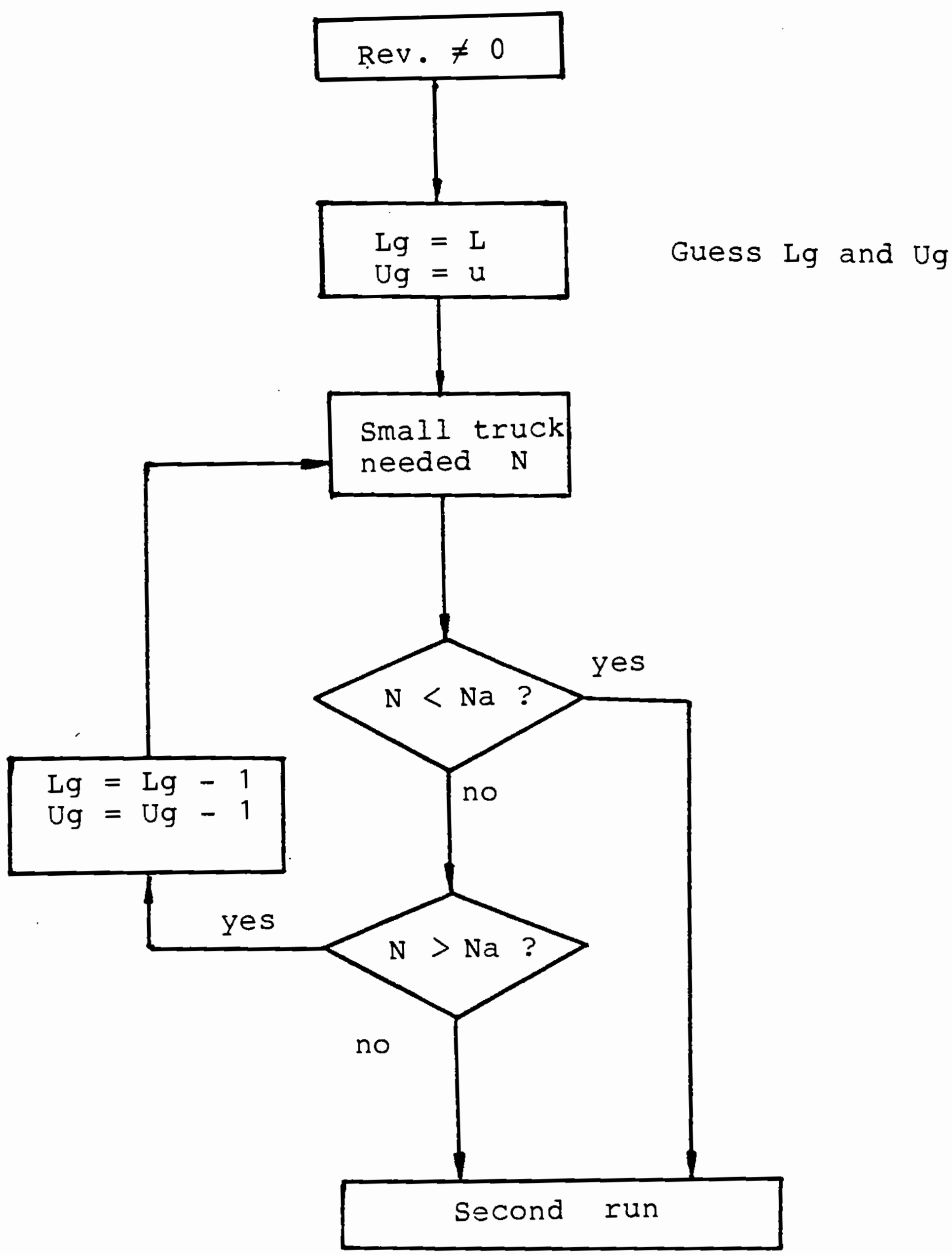

Figure 7. Flow chart for two truck types in category 2.

11. DATA COLLECTION AND DISCUSSION OF UNCERTAINTY

In the previous discussion, we assume loading rate, the minimum and maximum of trips of log trucks needed by every landing in a day, mean round trip traveling time to every landing, and total cost per trip for every landing are known. In practice, we need to collect data and estimate these inputs for our network model.

11.1. Data and Data Analysis

11.1.1 Data Analysis With Normality Assumption

Before a truck schedule can be determined the following data needs to be collected: (a) the maximum and minimum number of truck loads to be hauled from each landing in the day in question; (b) mean traveling time associated with each landing; (c) mean loading rate associated with each landing; (d) mean unloading capacity at the timber yard and (e) the crew working schedule of each landing and of the timber yard.

(1) Trips Needed By Each Landing

The landings can be divided into two groups. The first group has those landings with accumulated volume that must be transported by the end of the day so that there is enough room for the next day. The second group which has surplus space which permits some wood to be left for the next day.

Depending on the type of the group, the maximum and minimum trips needed by a landing can be found in different ways. For the first group, i.e., the group with volume that must be transported, we need to know the minimum and maximum volume that will be delivered in the next day from each landing. This is to say that we need to find a prediction interval for the next day's production.

Let Y_i be the daily production of a landing in the i-th day. If Y_i or $\log(Y_i)$ or other transformation of Y_i is approximately normally distributed, the $(1-\alpha)$ confidence level prediction interval is as follows:

$$\overline{Y} - t_\alpha * [S^2 * (n+1)/n]^{1/2} \leq Y^0 \leq \overline{Y} + t_\alpha * [S^2 * (n+1)/n]^{1/2}$$

where $\overline{Y} = \dfrac{\Sigma\ (Y_i)}{n}$, $S^2 = \dfrac{\Sigma\ (\ Y_i - \overline{Y}\ )^2}{n-1}$ and this can be easily converted into a prediction interval of trips.

For the second group, depending more on the user's opinion, the minimum amount of the volume which must be transported can be the lower limit of the prediction interval of the daily production of a landing or even a value that is smaller than the lower limit. While the value that is smaller than the lower limit. While the maximum amount of the tree length can be the sum of the daily production and the accumulated storage or less. The user can, according to a particular situation, decide the minimum amount and maximum amount to be transported. Then, the maximum and minimum daily trips needed by a landing for the next day can be determined.

(2) Loading and Unloading Rate

For the loading and unloading rate, we need to find the lower limit of the prediction interval. If we let Y_i be the loading rate (the number of trucks that can be loaded per hour) at the i-th hour, $\overline{Y}$ be the mean of the loading rate of a sample, S^2 be the sample variance and Y^0 be the future value of the loading rate, then we have

$$t = \frac{Y^0 - \overline{Y}}{[\ S^2 * (n+1)/n\ ]^{1/2}}$$

and $$P\left\{ \frac{Y^0 - \overline{Y}}{(\ S^2 * (n+1)/n\ )^{1/2}} \geq t_\alpha \right\} = 1-\alpha$$

Therefore, the lower limit of a $(1-\alpha)$ confidence level prediction interval for Y^0 is

$$\{\ t_\alpha * [\ S^2 * (n+1)/n\ ]^{1/2} + \overline{Y}\ \}$$

or $$Y^0 \geq t_\alpha * [\ S^2 * (n+1)/n\ ]^{1/2} + \overline{Y}$$

The unloading rate can be found in a similar way. The data collection , however, should be under a condition which there are trucks waiting in line. By unloading rate, we actually mean the minimum of the truck unloading rate or the production rate through the timber yard. Since the timber yard has limited storage space, the production process inside the timber yard has no space for inventory. Quite often, the unloading rate of the unloading machines in the timber yard is much greater than the production line and the unloading activity must wait until the production line is available to process more wood.

(3) Traveling Time

As previously defined, the traveling time, TM, consists of three components: the traveling time from the timber yard to the landing in question, the loading time and the return time to the timber yard from the landing.

$$Y = up + down + load = travel + load$$

where **up** is the time from the timber yard to the landing; **down** is the time from the landing to the timber yard; "load" is the loading time and **travel** is the round trip traveling time.

What we try to find here is the width of the prediction interval of the traveling time. It is a primary consideration in setting the width of the time interval for the network model.

Let Y_i be the i-th observation. Again, if we assume that Y_i or any kind of transformation of Y_i is approximately normally distributed, we can find the prediction interval of Y^0 as follows:

$$\{ \bar{Y} - t_\alpha * (S^2 * (n+1)/n)^{1/2}, \quad \bar{Y} + t_\alpha * (S^2 * (n+1)/n)^{1/2} \}$$

for which the confidence level is $(1-\alpha)$.

Where $\bar{Y} = \dfrac{\Sigma (Y_i)}{n}$ and $S^2 = \dfrac{\Sigma (Y_i - \bar{Y})^2}{n - 1}$

Now, let D be the difference between the upper limit and the lower limit of the confidence interval, i.e.,

$$D = \text{upper limit} - \text{lower limit}$$

$$= \{ \bar{Y} + t_\alpha * [S^2 * (n+1)/n]^{1/2} \} - \{ \bar{Y} - t_\alpha * [S^2 * (n+1)/n]^{1/2} \}$$

$$= 2 * t_\alpha * [S^2 (n+1)/n]^{1/2}$$

This suggests that the width of the prediction interval of Y^0 does not depend on the sample mean, $\bar{Y}$, but the sample variance of, S^2.

If the variances of the traveling time Y_i of all landings are not significantly different, then, the prediction interval width D can be assumed to be the same for all landings. Here, our hypothesis are:

NH: all variance σ^2 are equal for the **n** landings

AH: not all the variance are equal

If the variances of the traveling times associated with the landings are not the same, consequently, the widths of the prediction intervals of the traveling times will be different. In this case, we let

$$D = \max (D_1, D_2, \ldots , D_n)$$

where **n**, again, is the number of the landings.

11.1.2 Non-parametric Analysis

Without the assumption of a normal distributions, the nonparametric method can be used here. Let $Y_1, \ldots, Y_n$ be the random observations. We order the observations as follows:

$$Y_{(1)} < Y_{(2)} < \ldots < Y_{(n)}$$

the prediction interval of a future observation at $(1-\alpha)$ confidence level can be expressed as follows:

$$P [l \leq Y^0 \leq u] \geq 1 - \alpha$$

and $$P [l \leq y^0 \leq u] = E [F(Y_i) - F(Y_j)].$$

According to Hogg and Craig, A. T. (1971), we have:

$$E[F(Y_j) - F(Y_i)] = (j-i)/(n+1)$$

where i and j are chosen such that

$$(j-i)/(n+1) = 1-\alpha$$

Note that because i and j, and n are integers it might not be possible to attain exact confidence level $1-\alpha$.

For example, for 95% approximate confidence level, We could take

$$i = floor[.025*(n+1)]$$

and $$j = seiling[.975*(n+1)]$$

where floor(α) denote the largest integer $\leq \alpha$

seiling(α) denote the smallest integer $\geq \alpha$

11.2 Discussion of Uncertainty and Precision

11.2.1 Confidence Level At Which the Model Works

In the model formulation section, we connect the departure nodes and arrival nodes according to the traveling time associated with the corresponding landing. The width of the time interval is one hour. Obviously, if the traveling time spent by a particular truck in the future observation happens to be what we expect, the arrival time will fall in the time interval specified by the model. In other words, the truck does follow the arc and goes to the node which this arc points to.

If the traveling time, however, is shorter or longer than what we expect, the arrival time will fall in adjacent time interval instead of the one that we expect. This is what we try to avoid with the choice of the width of the time interval.

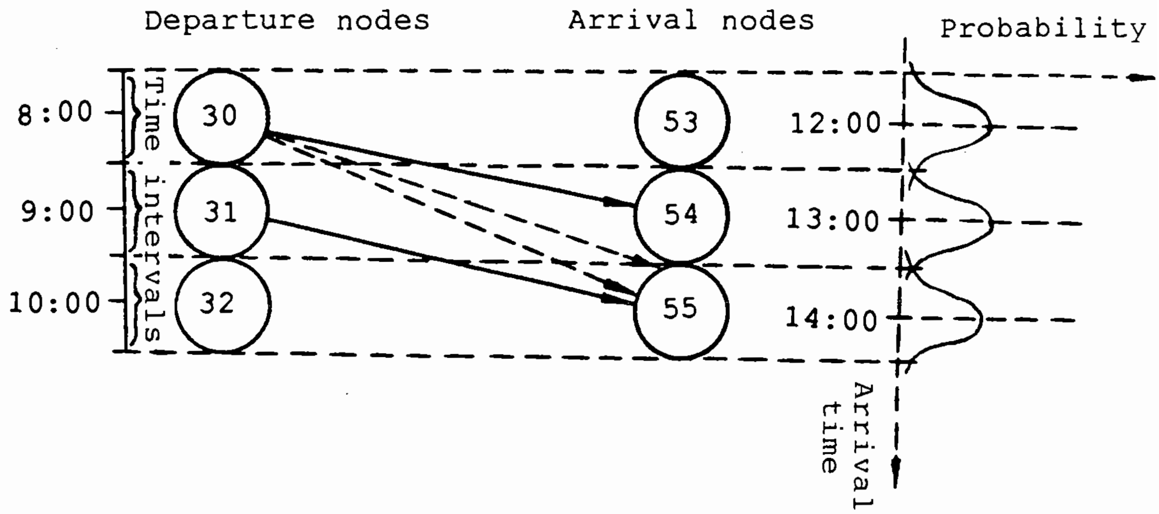

Figure 8. Illustration of uncertainty. Node 30,31 and 32 are departure nodes from a same landing. Node 53, 54 and 55 are arrival nodes. The traveling time is five hours. If the traveling time is longer than five hours by thirty minutes or more, the truck from node 30 will not go to node 54 but node 55.

As a numerical example(diagram 8), assume the following: the traveling time associated with landing i is 5 hours; the perticular departure time considered here is 8 o'clock; the width of the time interval is one hour; the departure node which represents 8 o'clock is node 30; the arrival node that represents (5+8) o'clock is node 54; According to the rule with which we formulated the model, we will have a arc from node 30 to node 54.

If the actual traveling time is 6 hours, the arrival time will be 14:00 instead of 13:00. This means that the flow from node 30 will not go to node 54, but node 55. If this happens too often, we say that the model fails to work. One may notice that in this example the model still works as long as the traveling time is not longer or shorter than the expected value by more than half hour.

In order to make sure that the model works, say, 95 times out 100, or at 95% confidence level, we have to choose a proper width for the time interval. Of course, if the width of the time interval that we choose is wider than that of the prediction interval of the traveling time, the arrival time will fall in our expected time interval at least 95% of the time. Since the width of the prediction interval of the traveling time has been discussed in the previous section, we will not repeat it here.

To summarize, we have to make three kinds of estimations before we can set up our model. They all affect the confidence level at which the model works but in different ways. If we set the confidence levels of the prediction intervals of the traveling time, of the daily production of the landings and of the loading and unloading rate equal to 95%, we find that the model will work at the 86 % confidence level.

11.2.2 Precision of the Model

According to the previous discussion, we wish the time interval as wide as possible so that a high confidence level can be obtained. It is, however, obvious that our results will be too rough if the time interval is too wide. We hope that we can control the truck arrivals as precisely as possible. In the previous example, the truck arrival time will fall in the one hour time interval centered at 13:00. In other words, we can control a given truck arrival at the timber yard at any time from 12:30 to 13:30. If the time interval is twenty minutes instead of one hour, however, we can arrange a truck arrival at timber yard either at any time between 13:00 and 13:20, or any time between 13:20 and 13:40 or 13:40 to 14:00. Therefore, we can control the precision by adjusting the width of the time interval.

If we want to simultaneously increase the confidence level at which the model works and while not decreasing precision, we have to improve the road condition, truck availability, communication equipment and so on. These can decrease the variance of the traveling time. If the variance of the traveling time is reduced while the width of the time interval is kept the same, the confidence level will be increased.

12. LITERATURE CITED

Barr, R., F. Glover, and D. Klingman. An Improved Version of the Out-Of-Kilter Method and A Comparitive Study of Computer Codes. Mathematical Programming 7(1):60-87.

Bellam, R., Esogbue, A. and I. Nabeshima. 1982. Mathematical Aspects of Scheduling and Applications. Pergamon Press. 329 p.

Hatch, R. 1975. Bench Marks Comparing Transportation Codes based on Primal Simplex and Primal-Dual Algorithms. Operations Research 23(6):1167-1171.

Hogg, R.V. & Craig, A. T. 1971. Introduction to Mathematical Statistics. The Macmillan Company, Collier-Macmillan Limited, London. 411 p.

Kennington, J. and R. Helgason. 1980. Algorithms for Network Programming. John Wiley and Sons. 291 p.

Levin, Amos. 1969. Some Fleet Routing and Scheduling Problems in Air Transportation Systems. Flight Transportation Laboratory. Mass. Inst. of Tech. 125 p.

Reece, R. M. 1974. The Marketing of A Forest Product: A Chance-Constrained Transportation Model. Studies in Marketing No. 20. The Univ. of Texas at Austin. 66 p.

Smith, D. K. 1982. Network Optimsation Practice. A Computational Guide. Ellis Horwood Limited. 237 p.

Wang, L. H. (1986). A Pilot Study on Dispatching Vehicles to Transport Tree length in HeiLongJiang Forest Area.

(unpublished). Northeastern Forestry University, Harbin, People's Republic of China. 74 p.

Wren, A., Editor. 1981 Computer Scheduling of Public Transport, North Holland Publishing. 359 p.

APPENDICES

APPENDIX A

THE CORRECTION OF THE COMPUTER CODES OF SMITH(1982)

a. A Brief Review of The "Out-Of-Kilter" Algorithm

A wide variety of network optimization problems belong to or can converted to following network problems defined on network (N, A):

$$\text{Min} \sum_{(i,j)\in A} (C_{ij} * X_{ij})$$

$$\text{S.T.} \sum_{(i,j)\in A} (X_{ij}) - \sum_{(k,j)\in A} (X_{kj}) = 0 \quad \text{for } i=1, \ldots , m$$

$$0 \le L_{ij} \le U_{ij} \quad \text{for all } (i,j) \in A$$

where N is the set of nodes and A is the set of arcs.

The "out-of-kilter" algorithm is related with the primal-dual algorithm. As pointed out by SHAPIRO (1979), "both primal-dual algorithm and the "out-of-kilter" rely on iterative use of the "maximal flow labeling algorithm". The ways by which they seek the optimal solution, however, are quite different. If we let πi be any dual solution which belongs to m dimensional space, R^m, it is optimal and its primal solution X(i,j) is also optimal if and only if

$$\sum_{(i,j)\in A}(X_{ij}) - \sum_{(k,j)\in A}(X_{kj}) = 0 \quad \text{for } i=1, \ldots , m \qquad (1)$$

$$C_{ij} + \pi_i - \pi_j < 0, \quad X_{ij} = U_{ij} \qquad (2)$$

$$C_{ij} + \pi_i - \pi_j = 0, \quad L_{ij} \le X_{ij} \le U_{ij} \qquad (3)$$

$$C_{ij} + \pi_i - \pi_j > 0, \quad X_{ij} = L_{ij} \tag{4}$$

The primal-dual algorithm maintains (2),(3),(4) at each iteration while trying to attain the feasibility condition. The "out-of-kilter" algorithm takes the opposite approach and maintains only (1) and seeks to attain the others.

The term ($C_{ij} + \pi_i - \pi_j$) may be thought of as a single variable (SMITH 1982). With this variable as the vertical axis, the flow as the horizontal axis, we can construct a coordinate system. A flow in an arc(i,j) corresponds to a point in this system. The point which satisfies optimal condition (2), (3), (4) will be on the solid line shown below:

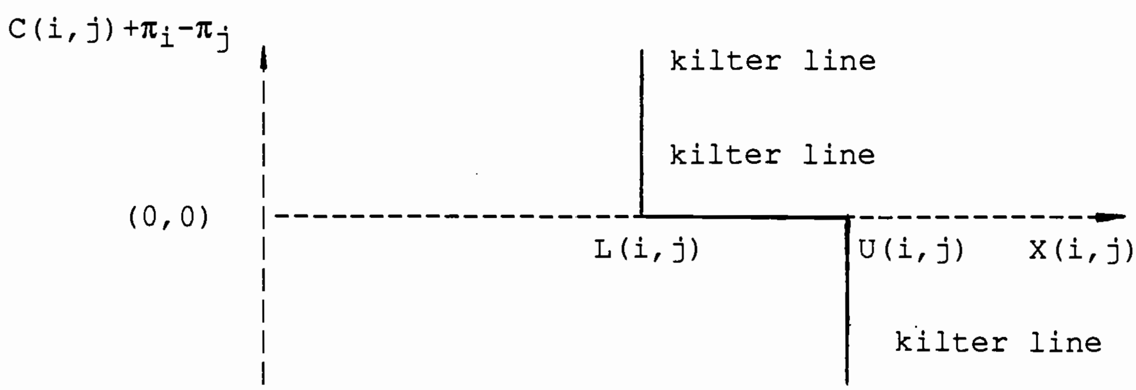

Figure 9. Kilter line.

This line is referred to as the "kilter line" for the arc (i,j). Therefore, if an arc (i,j) falls on the kilter line, it satisfies the optimal condition (2), (3), (4). Recall that the "out-of-kilter" algorithm maintains condition (1) all the time. Therefore, if the arc position in the diagram falls on the kilter line, it satisfies all optimal conditions. On the other hand, if it falls in another position, we call it "out of kilter". The"out-of kilter" algorithm trys to bring all arcs into kilter one by one.

b. Discussion of Smith Computer Code Correction

Smith (1982) provides a computer code to implement the "out-of-kilter" algorithm. Unfortunately, the code listing provided by Smith has two kinds of mistakes. One can be classified as a labeling algorithm mistake and the other as an "out-of-kilter" algorithm mistake. The details about the "out-of-kilter" algorithm can be found in many refrence books and articles (for example, SMITH 1982, R.J. CLASEN 1968 , E.P. DURBIN and D.M. KROENKE 1967)

In order to make it easy to follow, let's start from the beginning of the algorithm. In step 0 of computer program in SMITH (1982), the arbitrary multiplier πi to node i and flow X(i,j) to arc (i,j) will be assigned with their values being zeroes as an arbitrary feasible solution (corresponding optimal condition [1]). In step I, the arc(i,j) which is the first arc in the arc index will be picked up (note here that the arc picked can be any arbitrary arc). Then, the arc is classified. First, is it in kilter or out of kilter? If it is out of kilter, is it the arc which has to be set to (t,s) arc or (s,t). So far, the program is correct.

In step II and III, however, the program has mistakes. After an out of kilter arc (i,j) is found, an attempt is made to find a flow augmenting chain from **s** to **t** according to a labeling algorithm. If this can be done, the flow in this chain will be increased. By doing this, the out of kilter arc (i,j) either will be brought to the "kilter line" or close to it. In the later case, some other arc in the chain will be brought into the "kilter line". Unfortunately, some mistakes were made here. First, since we want to label source **s**. According to the labeling algorithm, the first part of the label is the flag which indicates the preceding node. Of course, if the arc in question is the (t,s) kind, we set **s** = J1 and **t** = I1. Therefore, the first part of the label of node **s** is a(s) = t, not a(s) = m1 which is arc order number in that program. If the arc in question is (s,t) kind, we set **s** = I1, **t** = J1. The first part of the label of node **s** is a(s) = - t, not a(s) = (-m1).

After the above procedure is compeleted, the arc list will be scanned through until an arc (i,j) for which one node was labeled and the other not, and

for a forward arc,

if $C(i,j) + \pi i - \pi j > 0$ and $X(i,j) < L(i,j)$

or $C(i,j) + \pi i - \pi j \leq 0$ and $X(i,j) < U(i,j)$

for a reverse arc,

if $C(i,j) + \pi i - \pi j \geq 0$ and $X(i,j) > L(i,j)$

or $C(i,j) + \pi i - \pi j < 0$ and $X(i,j) > U(i,j)$

is found. Since the scanning index or loop counter variable m4 used here is the order number of the arcs, the follow label method is incorrect:

A(J4)=m4 for forward arcs

A(I4)=-m4 for reverse arcs

because m4 is not the preceding node, it is the order number of the arc in question. In fact, it can be simply labeled like this:

A(J4) = I4 for forward arcs

A(J4) = - J4 for reverse arcs

This process is repeated until finally **t** labeled and the flow augementing chain found or no more arcs can be labeled.

If it was labeled, like we said before, the flow in this chain will be increased. To do so, we read back the labels and find arcs in the chain one by one, and increase the flow in each arc. The increase in flow will be the largest possible flow which does not move any arcs in the chain farther from the kilter line. Smith (1982) did not do this successfully. In this paper, we read back the labels by reading the first part of the label of node **t**. This provides the preceding node of this node in the chain. With the start node and the end node known, the arc file is scanned in order to find the order number of the arc. Then the flow is increased in it. This process is repeated until the sink node **t** is reached. Then, all labels are discarded and we go back to the step I to find another out of kilter arc.

In the labeling process, if it is not possible to find a complete flow augmenting chain, then step III begins.

For forward arcs (or those with node i labeled and j not) for which $C(i,j) + \pi i - \pi j > 0$ and $L(i,j) \leq X(i,j) < U(i,j)$, find $\delta 1$ as the smallest value of $C(i,j) + \pi i - \pi j$, i.e.,

$$\delta 1 = \min \; (C(i,j) + \pi i - \pi j)$$

For reverse arcs (or the arcs for which node j labeled and i not), and $C(i,j) + \pi i - \pi j < 0$ and $L(i,j) <$ $X(i,j) \quad U(i,j)$, find

$$\delta 2 = \min \; | C(i,j) + \pi i - \pi j |$$

Then, find

$$\delta = \min \; (\delta 1, \; \delta 2)$$

and add δ to the π-multipliers of all unlabeled nodes.

The mistake here in Smith (1982) is as follows: both the forward arc and the reverse arc, as long as it satisfies $L(i,j) < X(i,j) \leq U(i,j)$, will be taken into consideration to determine $\delta 1$. As we stated before, only the reverse arc which satisfies both

$$L(i,j) < X(i,j) \leq U(i,j)$$

and $C(i,j) + \pi i - \pi j < 0$

should be considered to determine $\delta 1$. The forward arc is not an eligible candidate. This is because it can be included in our flow augmenting chain and be brought into or close to kilter by increasing flow in it if it is forward arc and also satisfies $C(i,j) + \pi i - \pi j < 0$. Therefore, it is not the kind of arc which should be used to determine $\delta 1$.

APPENDIX B

NETWORK FILE AND OUTPUT

Table 18. Network file of example 3.

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
1	1	2	3	4	13
2	1	3	4	5	13
3	1	4	2	3	23
4	1	5	2	3	23
5	1	6	5	6	48
6	1	7	5	6	48
7	1	8	1	2	51
8	1	9	1	2	51
9	1	10	2	3	51
10	1	11	4	5	37
11	2	12	0	39	9999
12	3	12	0	39	9999
13	4	12	0	39	9999
14	5	12	0	39	9999
15	6	12	0	39	9999
16	7	12	0	39	9999
17	8	12	0	39	9999
18	9	12	0	39	9999
19	10	12	0	39	9999
20	11	12	0	39	9999
21	2	13	0	2	0
22	2	14	0	2	0
23	2	15	0	2	0
24	2	16	0	2	0
25	2	17	0	2	0
26	2	18	0	2	0
27	2	19	0	2	0
28	2	20	0	2	0
29	2	21	0	2	0
30	2	22	0	2	0
31	2	23	0	2	0
32	2	24	0	2	0
33	2	25	0	2	0
34	2	26	0	2	0
35	2	27	0	2	0
36	2	28	0	2	0
37	2	29	0	2	0
38	2	30	0	2	0
39	2	31	0	2	0
40	2	32	0	2	0
41	2	33	0	2	0
42	2	34	0	2	0
43	2	35	0	2	0
44	2	36	0	2	0
45	3	37	0	2	0

Table 18 (cont.) Network file of example 3

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
46	3	38	0	2	0
47	3	39	0	2	0
48	3	40	0	2	0
49	3	41	0	2	0
50	3	42	0	2	0
51	3	43	0	2	0
52	3	44	0	2	0
53	3	45	0	2	0
54	3	46	0	2	0
55	3	47	0	2	0
56	3	48	0	2	0
57	3	49	0	2	0
58	3	50	0	2	0
59	3	51	0	2	0
60	3	52	0	2	0
61	3	53	0	2	0
62	3	54	0	2	0
63	3	55	0	2	0
64	3	56	0	2	0
65	3	57	0	2	0
66	3	58	0	2	0
67	3	59	0	2	0
68	3	60	0	2	0
69	4	61	0	2	0
70	4	62	0	2	0
71	4	63	0	2	0
72	4	64	0	2	0
73	4	65	0	2	0
74	4	66	0	2	0
75	4	67	0	2	0
76	4	68	0	2	0
77	4	69	0	2	0
78	4	70	0	2	0
79	4	71	0	2	0
80	4	72	0	2	0
81	4	73	0	2	0
82	4	74	0	2	0
83	4	75	0	2	0
84	4	76	0	2	0
85	4	77	0	2	0
86	4	78	0	2	0
87	4	79	0	2	0
88	4	80	0	2	0
89	4	81	0	2	0
90	4	82	0	2	0

Table 18 (cont.) Network file of example 3

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
91	4	83	0	2	0
92	4	84	0	2	0
93	5	85	0	2	0
94	5	86	0	2	0
95	5	87	0	2	0
96	5	88	0	2	0
97	5	89	0	2	0
98	5	90	0	2	0
99	5	91	0	2	0
100	5	92	0	2	0
101	5	93	0	2	0
102	5	94	0	2	0
103	5	95	0	2	0
104	5	96	0	2	0
105	5	97	0	2	0
106	5	98	0	2	0
107	5	99	0	2	0
108	5	100	0	2	0
109	5	101	0	2	0
110	5	102	0	2	0
111	5	103	0	2	0
112	5	104	0	2	0
113	5	105	0	2	0
114	5	106	0	2	0
115	5	107	0	2	0
116	5	108	0	2	0
117	6	109	0	2	0
118	6	110	0	2	0
119	6	111	0	2	0
120	6	112	0	2	0
121	6	113	0	2	0
122	6	114	0	2	0
123	6	115	0	2	0
124	6	116	0	2	0
125	6	117	0	2	0
126	6	118	0	2	0
127	6	119	0	2	0
128	6	120	0	2	0
129	6	121	0	2	0
130	6	122	0	2	0
131	6	123	0	2	0
132	6	124	0	2	0
133	6	125	0	2	0
134	6	126	0	2	0
135	6	127	0	2	0

Table 18 (cont.) Network file of example 3

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
136	6	128	0	2	0
137	6	129	0	2	0
138	6	130	0	2	0
139	6	131	0	2	0
140	6	132	0	2	0
141	7	133	0	2	0
142	7	134	0	2	0
143	7	135	0	2	0
144	7	136	0	2	0
145	7	137	0	2	0
146	7	138	0	2	0
147	7	139	0	2	0
148	7	140	0	2	0
149	7	141	0	2	0
150	7	142	0	2	0
151	7	143	0	2	0
152	7	144	0	2	0
153	7	145	0	2	0
154	7	146	0	2	0
155	7	147	0	2	0
156	7	148	0	2	0
157	7	149	0	2	0
158	7	150	0	2	0
159	7	151	0	2	0
160	7	152	0	2	0
161	7	153	0	2	0
162	7	154	0	2	0
163	7	155	0	2	0
164	7	156	0	2	0
165	8	157	0	2	0
166	8	158	0	2	0
167	8	159	0	2	0
168	8	160	0	2	0
169	8	161	0	2	0
170	8	162	0	2	0
171	8	163	0	2	0
172	8	164	0	2	0
173	8	165	0	2	0
174	8	166	0	2	0
175	8	167	0	2	0
176	8	168	0	2	0
177	8	169	0	2	0
178	8	170	0	2	0
179	8	171	0	2	0
180	8	172	0	2	0

Table 18 (cont.) Network file of example 3

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
181	8	173	0	2	0
182	8	174	0	2	0
183	8	175	0	2	0
184	8	176	0	2	0
185	8	177	0	2	0
186	8	178	0	2	0
187	8	179	0	2	0
188	8	180	0	2	0
189	9	181	0	2	0
190	9	182	0	2	0
191	9	183	0	2	0
192	9	184	0	2	0
193	9	185	0	2	0
194	9	186	0	2	0
195	9	187	0	2	0
196	9	188	0	2	0
197	9	189	0	2	0
198	9	190	0	2	0
199	9	191	0	2	0
200	9	192	0	2	0
201	9	193	0	2	0
202	9	194	0	2	0
203	9	195	0	2	0
204	9	196	0	2	0
205	9	197	0	2	0
206	9	198	0	2	0
207	9	199	0	2	0
208	9	200	0	2	0
209	9	201	0	2	0
210	9	202	0	2	0
211	9	203	0	2	0
212	9	204	0	2	0
213	10	205	0	2	0
214	10	206	0	2	0
215	10	207	0	2	0
216	10	208	0	2	0
217	10	209	0	2	0
218	10	210	0	2	0
219	10	211	0	2	0
220	10	212	0	2	0
221	10	213	0	2	0
222	10	214	0	2	0
223	10	215	0	2	0
224	10	216	0	2	0
225	10	217	0	2	0

Table 18 (cont.) Network file of example 3

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
226	10	218	0	2	0
227	10	219	0	2	0
228	10	220	0	2	0
229	10	221	0	2	0
230	10	222	0	2	0
231	10	223	0	2	0
232	10	224	0	2	0
233	10	225	0	2	0
234	10	226	0	2	0
235	10	227	0	2	0
236	10	228	0	2	0
237	11	229	0	2	0
238	11	230	0	2	0
239	11	231	0	2	0
240	11	232	0	2	0
241	11	233	0	2	0
242	11	234	0	2	0
243	11	235	0	2	0
244	11	236	0	2	0
245	11	237	0	2	0
246	11	238	0	2	0
247	11	239	0	2	0
248	11	240	0	2	0
249	11	241	0	2	0
250	11	242	0	2	0
251	11	243	0	2	0
252	11	244	0	2	0
253	11	245	0	2	0
254	11	246	0	2	0
255	11	247	0	2	0
256	11	248	0	2	0
257	11	249	0	2	0
258	11	250	0	2	0
259	11	251	0	2	0
260	11	252	0	2	0
261	13	255	0	39	0
262	14	256	0	39	0
263	15	257	0	39	0
264	16	258	0	39	0
265	17	259	0	39	0
266	18	260	0	39	0
267	19	261	0	39	0
268	20	262	0	39	0
269	21	263	0	39	0
270	22	264	0	39	0

Table 18 (cont.) Network file of example 3.

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
271	23	265	0	39	0
272	24	266	0	39	0
273	25	267	0	39	0
274	26	268	0	39	0
275	27	269	0	39	0
276	28	270	0	39	0
277	29	271	0	39	0
278	30	272	0	39	0
279	31	273	0	39	0
280	32	274	0	39	0
281	33	275	0	39	0
282	34	276	0	39	0
283	35	253	0	39	0
284	36	254	0	39	0
285	37	255	0	39	0
286	38	256	0	39	0
287	39	257	0	39	0
288	40	258	0	39	0
289	41	259	0	39	0
290	42	260	0	39	0
291	43	261	0	39	0
292	44	262	0	39	0
293	45	263	0	39	0
294	46	264	0	39	0
295	47	265	0	39	0
296	48	266	0	39	0
297	49	267	0	39	0
298	50	268	0	39	0
299	51	269	0	39	0
300	52	270	0	39	0
301	53	271	0	39	0
302	54	272	0	39	0
303	55	273	0	39	0
304	56	274	0	39	0
305	57	275	0	39	0
306	58	276	0	39	0
307	59	253	0	39	0
308	60	254	0	39	0
309	61	256	0	39	0
310	62	257	0	39	0
311	63	258	0	39	0
312	64	259	0	39	0
313	65	260	0	39	0
314	66	261	0	39	0
315	67	262	0	39	0

Table 18 (cont.) Network file of example 3

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
316	68	263	0	39	0
317	69	264	0	39	0
318	70	265	0	39	0
319	71	266	0	39	0
320	72	267	0	39	0
321	73	268	0	39	0
322	74	269	0	39	0
323	75	270	0	39	0
324	76	271	0	39	0
325	77	272	0	39	0
326	78	273	0	39	0
327	79	274	0	39	0
328	80	275	0	39	0
329	81	276	0	39	0
330	82	253	0	39	0
331	83	254	0	39	0
332	84	255	0	39	0
333	85	256	0	39	0
334	86	257	0	39	0
335	87	258	0	39	0
336	88	259	0	39	0
337	89	260	0	39	0
338	90	261	0	39	0
339	91	262	0	39	0
340	92	263	0	39	0
341	93	264	0	39	0
342	94	265	0	39	0
343	95	266	0	39	0
344	96	267	0	39	0
345	97	268	0	39	0
346	98	269	0	39	0
347	99	270	0	39	0
348	100	271	0	39	0
349	101	272	0	39	0
350	102	273	0	39	0
351	103	274	0	39	0
352	104	275	0	39	0
353	105	276	0	39	0
354	106	253	0	39	0
355	107	254	0	39	0
356	108	255	0	39	0
357	109	257	0	39	0
358	110	258	0	39	0
359	111	259	0	39	0
360	112	260	0	39	0

Table 18 (cont.) Network file of example 3

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
361	113	261	0	39	0
362	114	262	0	39	0
363	115	263	0	39	0
364	116	264	0	39	0
365	117	265	0	39	0
366	118	266	0	39	0
367	119	267	0	39	0
368	120	268	0	39	0
369	121	269	0	39	0
370	122	270	0	39	0
371	123	271	0	39	0
372	124	272	0	39	0
373	125	273	0	39	0
374	126	274	0	39	0
375	127	275	0	39	0
376	128	276	0	39	0
377	129	253	0	39	0
378	130	254	0	39	0
379	131	255	0	39	0
380	132	256	0	39	0
381	133	257	0	39	0
382	134	258	0	39	0
383	135	259	0	39	0
384	136	260	0	39	0
385	137	261	0	39	0
386	138	262	0	39	0
387	139	263	0	39	0
388	140	264	0	39	0
389	141	265	0	39	0
390	142	266	0	39	0
391	143	267	0	39	0
392	144	268	0	39	0
393	145	269	0	39	0
394	146	270	0	39	0
395	147	271	0	39	0
396	148	272	0	39	0
397	149	273	0	39	0
398	150	274	0	39	0
399	151	275	0	39	0
400	152	276	0	39	0
401	153	253	0	39	0
402	154	254	0	39	0
403	155	255	0	39	0
404	156	256	0	39	0
405	157	258	0	39	0

Table 18 (cont.) Network file of example 3

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
406	158	259	0	39	0
407	159	260	0	39	0
408	160	261	0	39	0
409	161	262	0	39	0
410	162	263	0	39	0
411	163	264	0	39	0
412	164	265	0	39	0
413	165	266	0	39	0
414	166	267	0	39	0
415	167	268	0	39	0
416	168	269	0	39	0
417	169	270	0	39	0
418	170	271	0	39	0
419	171	272	0	39	0
420	172	273	0	39	0
421	173	274	0	39	0
422	174	275	0	39	0
423	175	276	0	39	0
424	176	253	0	39	0
425	177	254	0	39	0
426	178	255	0	39	0
427	179	256	0	39	0
428	180	257	0	39	0
429	181	258	0	39	0
430	182	259	0	39	0
431	183	260	0	39	0
432	184	261	0	39	0
433	185	262	0	39	0
434	186	263	0	39	0
435	187	264	0	39	0
436	188	265	0	39	0
437	189	266	0	39	0
438	190	267	0	39	0
439	191	268	0	39	0
440	192	269	0	39	0
441	193	270	0	39	0
442	194	271	0	39	0
443	195	272	0	39	0
444	196	273	0	39	0
445	197	274	0	39	0
446	198	275	0	39	0
447	199	276	0	39	0
448	200	253	0	39	0
449	201	254	0	39	0
450	202	255	0	39	0

Table 18 (cont.) Network file of example 3

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
451	203	256	0	39	0
452	204	257	0	39	0
453	205	255	0	39	0
454	206	256	0	39	0
455	207	257	0	39	0
456	208	258	0	39	0
457	209	259	0	39	0
458	210	260	0	39	0
459	211	261	0	39	0
460	212	262	0	39	0
461	213	263	0	39	0
462	214	264	0	39	0
463	215	265	0	39	0
464	216	266	0	39	0
465	217	267	0	39	0
466	218	268	0	39	0
467	219	269	0	39	0
468	220	270	0	39	0
469	221	271	0	39	0
470	222	272	0	39	0
471	223	273	0	39	0
472	224	274	0	39	0
473	225	275	0	39	0
474	226	276	0	39	0
475	227	253	0	39	0
476	228	254	0	39	0
477	229	258	0	39	0
478	230	259	0	39	0
479	231	260	0	39	0
480	232	261	0	39	0
481	233	262	0	39	0
482	234	263	0	39	0
483	235	264	0	39	0
484	236	265	0	39	0
485	237	266	0	39	0
486	238	267	0	39	0
487	239	268	0	39	0
488	240	269	0	39	0
489	241	270	0	39	0
490	242	271	0	39	0

Table 18 (cont.) Network file of example 3

Arc index	Start nodes	End nodes	Low bounds	Upper bounds	Cost ($)
491	243	272	0	39	0
492	244	273	0	39	0
493	245	274	0	39	0
494	246	275	0	39	0
495	247	276	0	39	0
496	248	253	0	39	0
497	249	254	0	39	0
498	250	255	0	39	0
499	251	256	0	39	0
500	252	257	0	39	0
501	253	277	0	0	-60
502	254	277	0	0	-60
503	255	277	0	0	-60
504	256	277	0	0	-60
505	257	277	0	0	-60
506	258	277	0	0	-60
507	259	277	0	0	-60
508	260	277	0	0	-60
509	261	277	0	4	-60
510	262	277	0	4	-60
511	263	277	0	4	-60
512	264	277	0	4	-60
513	265	277	0	4	-60
514	266	277	0	4	-60
515	267	277	0	4	-60
516	268	277	0	4	-60
517	269	277	0	4	-60
518	270	277	0	4	-60
519	271	277	0	4	-60
520	272	277	0	4	-60
521	273	277	0	4	-60
522	274	277	0	4	-60
523	275	277	0	4	-60
524	276	277	0	4	-60
525	277	12	0	39	0
526	12	1	0	39	0

Table 19. Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
1	3	4	4	1	2
2	4	5	5	1	3
3	2	3	3	1	4
4	2	3	3	1	5
5	5	6	6	1	6
6	5	6	6	1	7
7	1	2	2	1	8
8	1	2	2	1	9
9	2	3	3	1	10
10	4	5	5	1	11
11	0	39	0	2	12
12	0	39	0	3	12
13	0	39	0	4	12
14	0	39	0	5	12
15	0	39	0	6	12
16	0	39	0	7	12
17	0	39	0	8	12
18	0	39	0	9	12
19	0	39	0	10	12
20	0	39	0	11	12
21	0	2	0	2	13
22	0	2	0	2	14
23	0	2	0	2	15
24	0	2	0	2	16
25	0	2	0	2	17
26	0	2	0	2	18
27	0	2	2	2	19
28	0	2	1	2	20
29	0	2	0	2	21
30	0	2	0	2	22
31	0	2	0	2	23
32	0	2	0	2	24
33	0	2	1	2	25
34	0	2	0	2	26
35	0	2	0	2	27
36	0	2	0	2	28
37	0	2	0	2	29
38	0	2	0	2	30
39	0	2	0	2	31
40	0	2	0	2	32
41	0	2	0	2	33
42	0	2	0	2	34
43	0	2	0	2	35
44	0	2	0	2	36
45	0	2	0	3	37

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
46	0	2	0	3	38
47	0	2	0	3	39
48	0	2	0	3	40
49	0	2	0	3	41
50	0	2	0	3	42
51	0	2	2	3	43
52	0	2	2	3	44
53	0	2	0	3	45
54	0	2	0	3	46
55	0	2	0	3	47
56	0	2	0	3	48
57	0	2	0	3	49
58	0	2	1	3	50
59	0	2	0	3	51
60	0	2	0	3	52
61	0	2	0	3	53
62	0	2	0	3	54
63	0	2	0	3	55
64	0	2	0	3	56
65	0	2	0	3	57
66	0	2	0	3	58
67	0	2	0	3	59
68	0	2	0	3	60
69	0	2	0	4	61
70	0	2	0	4	62
71	0	2	0	4	63
72	0	2	0	4	64
73	0	2	0	4	65
74	0	2	0	4	66
75	0	2	1	4	67
76	0	2	1	4	68
77	0	2	0	4	69
78	0	2	0	4	70
79	0	2	0	4	71
80	0	2	0	4	72
81	0	2	1	4	73
82	0	2	0	4	74
83	0	2	0	4	75
84	0	2	0	4	76
85	0	2	0	4	77
86	0	2	0	4	78
87	0	2	0	4	79
88	0	2	0	4	80
89	0	2	0	4	81
90	0	2	0	4	82

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
91	0	2	0	4	83
92	0	2	0	4	84
93	0	2	0	5	85
94	0	2	0	5	86
95	0	2	0	5	87
96	0	2	0	5	88
97	0	2	0	5	89
98	0	2	0	5	90
99	0	2	0	5	91
100	0	2	2	5	92
101	0	2	0	5	93
102	0	2	0	5	94
103	0	2	0	5	95
104	0	2	0	5	96
105	0	2	0	5	97
106	0	2	1	5	98
107	0	2	0	5	99
108	0	2	0	5	100
109	0	2	0	5	101
110	0	2	0	5	102
111	0	2	0	5	103
112	0	2	0	5	104
113	0	2	0	5	105
114	0	2	0	5	106
115	0	2	0	5	107
116	0	2	0	5	108
117	0	2	0	6	109
118	0	2	0	6	110
119	0	2	0	6	111
120	0	2	0	6	112
121	0	2	0	6	113
122	0	2	0	6	114
123	0	2	1	6	115
124	0	2	2	6	116
125	0	2	2	6	117
126	0	2	0	6	118
127	0	2	0	6	119
128	0	2	0	6	120
129	0	2	1	6	121
130	0	2	0	6	122
131	0	2	0	6	123
132	0	2	0	6	124
133	0	2	0	6	125
134	0	2	0	6	126
135	0	2	0	6	127

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
136	0	2	0	6	128
137	0	2	0	6	129
138	0	2	0	6	130
139	0	2	0	6	131
140	0	2	0	6	132
141	0	2	0	7	133
142	0	2	0	7	134
143	0	2	0	7	135
144	0	2	0	7	136
145	0	2	0	7	137
146	0	2	0	7	138
147	0	2	0	7	139
148	0	2	2	7	140
149	0	2	2	7	141
150	0	2	1	7	142
151	0	2	0	7	143
152	0	2	0	7	144
153	0	2	1	7	145
154	0	2	0	7	146
155	0	2	0	7	147
156	0	2	0	7	148
157	0	2	0	7	149
158	0	2	0	7	150
159	0	2	0	7	151
160	0	2	0	7	152
161	0	2	0	7	153
162	0	2	0	7	154
163	0	2	0	7	155
164	0	2	0	7	156
165	0	2	0	8	157
166	0	2	0	8	158
167	0	2	0	8	159
168	0	2	0	8	160
169	0	2	0	8	161
170	0	2	0	8	162
171	0	2	0	8	163
172	0	2	0	8	164
173	0	2	1	8	165
174	0	2	0	8	166
175	0	2	0	8	167
176	0	2	0	8	168
177	0	2	1	8	169
178	0	2	0	8	170
179	0	2	0	8	171
180	0	2	0	8	172

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
181	0	2	0	8	173
182	0	2	0	8	174
183	0	2	0	8	175
184	0	2	0	8	176
185	0	2	0	8	177
186	0	2	0	8	178
187	0	2	0	8	179
188	0	2	0	8	180
189	0	2	0	9	181
190	0	2	0	9	182
191	0	2	0	9	183
192	0	2	0	9	184
193	0	2	0	9	185
194	0	2	0	9	186
195	0	2	0	9	187
196	0	2	0	9	188
197	0	2	1	9	189
198	0	2	0	9	190
199	0	2	0	9	191
200	0	2	0	9	192
201	0	2	1	9	193
202	0	2	0	9	194
203	0	2	0	9	195
204	0	2	0	9	196
205	0	2	0	9	197
206	0	2	0	9	198
207	0	2	0	9	199
208	0	2	0	9	200
209	0	2	0	9	201
210	0	2	0	9	202
211	0	2	0	9	203
212	0	2	0	9	204
213	0	2	0	10	205
214	0	2	0	10	206
215	0	2	0	10	207
216	0	2	0	10	208
217	0	2	0	10	209
218	0	2	0	10	210
219	0	2	0	10	211
220	0	2	0	10	212
221	0	2	0	10	213
222	0	2	0	10	214
223	0	2	0	10	215
224	0	2	1	10	216
225	0	2	1	10	217

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
226	0	2	0	10	218
227	0	2	0	10	219
228	0	2	1	10	220
229	0	2	0	10	221
230	0	2	0	10	222
231	0	2	0	10	223
232	0	2	0	10	224
233	0	2	0	10	225
234	0	2	0	10	226
235	0	2	0	10	227
236	0	2	0	10	228
237	0	2	0	11	229
238	0	2	0	11	230
239	0	2	0	11	231
240	0	2	0	11	232
241	0	2	0	11	233
242	0	2	0	11	234
243	0	2	0	11	235
244	0	2	0	11	236
245	0	2	0	11	237
246	0	2	2	11	238
247	0	2	2	11	239
248	0	2	1	11	240
249	0	2	0	11	241
250	0	2	0	11	242
251	0	2	0	11	243
252	0	2	0	11	244
253	0	2	0	11	245
254	0	2	0	11	246
255	0	2	0	11	247
256	0	2	0	11	248
257	0	2	0	11	249
258	0	2	0	11	250
259	0	2	0	11	251
260	0	2	0	11	252
261	0	39	0	13	255
262	0	39	0	14	256
263	0	39	0	15	257
264	0	39	0	16	258
265	0	39	0	17	259
266	0	39	0	18	260
267	0	39	2	19	261
268	0	39	1	20	262
269	0	39	0	21	263
270	0	39	0	22	264

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
271	0	39	0	23	265
272	0	39	0	24	266
273	0	39	1	25	267
274	0	39	0	26	268
275	0	39	0	27	269
276	0	39	0	28	270
277	0	39	0	29	271
278	0	39	0	30	272
279	0	39	0	31	273
280	0	39	0	32	274
281	0	39	0	33	275
282	0	39	0	34	276
283	0	39	0	35	253
284	0	39	0	36	254
285	0	39	0	37	255
286	0	39	0	38	256
287	0	39	0	39	257
288	0	39	0	40	258
289	0	39	0	41	259
290	0	39	0	42	260
291	0	39	2	43	261
292	0	39	2	44	262
293	0	39	0	45	263
294	0	39	0	46	264
295	0	39	0	47	265
296	0	39	0	48	266
297	0	39	0	49	267
298	0	39	1	50	268
299	0	39	0	51	269
300	0	39	0	52	270
301	0	39	0	53	271
302	0	39	0	54	272
303	0	39	0	55	273
304	0	39	0	56	274
305	0	39	0	57	275
306	0	39	0	58	276
307	0	39	0	59	253
308	0	39	0	60	254
309	0	39	0	61	256
310	0	39	0	62	257
311	0	39	0	63	258
312	0	39	0	64	259
313	0	39	0	65	260
314	0	39	0	66	261
315	0	39	1	67	262

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
316	0	39	1	68	263
317	0	39	0	69	264
318	0	39	0	70	265
319	0	39	0	71	266
320	0	39	0	72	267
321	0	39	1	73	268
322	0	39	0	74	269
323	0	39	0	75	270
324	0	39	0	76	271
325	0	39	0	77	272
326	0	39	0	78	273
327	0	39	0	79	274
328	0	39	0	80	275
329	0	39	0	81	276
330	0	39	0	82	253
331	0	39	0	83	254
332	0	39	0	84	255
333	0	39	0	85	256
334	0	39	0	86	257
335	0	39	0	87	258
336	0	39	0	88	259
337	0	39	0	89	260
338	0	39	0	90	261
339	0	39	0	91	262
340	0	39	2	92	263
341	0	39	0	93	264
342	0	39	0	94	265
343	0	39	0	95	266
344	0	39	0	96	267
345	0	39	0	97	268
346	0	39	1	98	269
347	0	39	0	99	270
348	0	39	0	100	271
349	0	39	0	101	272
350	0	39	0	102	273
351	0	39	0	103	274
352	0	39	0	104	275
353	0	39	0	105	276
354	0	39	0	106	253
355	0	39	0	107	254
356	0	39	0	108	255
357	0	39	0	109	257
358	0	39	0	110	258
359	0	39	0	111	259
360	0	39	0	112	260

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
361	0	39	0	113	261
362	0	39	0	114	262
363	0	39	1	115	263
364	0	39	2	116	264
365	0	39	2	117	265
366	0	39	0	118	266
367	0	39	0	119	267
368	0	39	0	120	268
369	0	39	1	121	269
370	0	39	0	122	270
371	0	39	0	123	271
372	0	39	0	124	272
373	0	39	0	125	273
374	0	39	0	126	274
375	0	39	0	127	275
376	0	39	0	128	276
377	0	39	0	129	253
378	0	39	0	130	254
379	0	39	0	131	255
380	0	39	0	132	256
381	0	39	0	133	257
382	0	39	0	134	258
383	0	39	0	135	259
384	0	39	0	136	260
385	0	39	0	137	261
386	0	39	0	138	262
387	0	39	0	139	263
388	0	39	2	140	264
389	0	39	2	141	265
390	0	39	1	142	266
391	0	39	0	143	267
392	0	39	0	144	268
393	0	39	1	145	269
394	0	39	0	146	270
395	0	39	0	147	271
396	0	39	0	148	272
397	0	39	0	149	273
398	0	39	0	150	274
399	0	39	0	151	275
400	0	39	0	152	276
401	0	39	0	153	253
402	0	39	0	154	254
403	0	39	0	155	255
404	0	39	0	156	256
405	0	39	0	157	258

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
406	0	39	0	158	259
407	0	39	0	159	260
408	0	39	0	160	261
409	0	39	0	161	262
410	0	39	0	162	263
411	0	39	0	163	264
412	0	39	0	164	265
413	0	39	1	165	266
414	0	39	0	166	267
415	0	39	0	167	268
416	0	39	0	168	269
417	0	39	1	169	270
418	0	39	0	170	271
419	0	39	0	171	272
420	0	39	0	172	273
421	0	39	0	173	274
422	0	39	0	174	275
423	0	39	0	175	276
424	0	39	0	176	253
425	0	39	0	177	254
426	0	39	0	178	255
427	0	39	0	179	256
428	0	39	0	180	257
429	0	39	0	181	258
430	0	39	0	182	259
431	0	39	0	183	260
432	0	39	0	184	261
433	0	39	0	185	262
434	0	39	0	186	263
435	0	39	0	187	264
436	0	39	0	188	265
437	0	39	1	189	266
438	0	39	0	190	267
439	0	39	0	191	268
440	0	39	0	192	269
441	0	39	1	193	270
442	0	39	0	194	271
443	0	39	0	195	272
444	0	39	0	196	273
445	0	39	0	197	274
446	0	39	0	198	275
447	0	39	0	199	276
448	0	39	0	200	253
449	0	39	0	201	254
450	0	39	0	202	255

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
451	0	39	0	203	256
452	0	39	0	204	257
453	0	39	0	205	255
454	0	39	0	206	256
455	0	39	0	207	257
456	0	39	0	208	258
457	0	39	0	209	259
458	0	39	0	210	260
459	0	39	0	211	261
460	0	39	0	212	262
461	0	39	0	213	263
462	0	39	0	214	264
463	0	39	0	215	265
464	0	39	1	216	266
465	0	39	1	217	267
466	0	39	0	218	268
467	0	39	0	219	269
468	0	39	1	220	270
469	0	39	0	221	271
470	0	39	0	222	272
471	0	39	0	223	273
472	0	39	0	224	274
473	0	39	0	225	275
474	0	39	0	226	276
475	0	39	0	227	253
476	0	39	0	228	254
477	0	39	0	229	258
478	0	39	0	230	259
479	0	39	0	231	260
480	0	39	0	232	261
481	0	39	0	233	262
482	0	39	0	234	263
483	0	39	0	235	264
484	0	39	0	236	265
485	0	39	0	237	266
486	0	39	2	238	267
487	0	39	2	239	268
488	0	39	1	240	269
489	0	39	0	241	270
490	0	39	0	242	271

Table 19 (cont.) Full output of the network of example 3.

Arc index	Low bounds	upper bounds	Flow in arcs	Start nodes	End nodes
491	0	39	0	243	272
492	0	39	0	244	273
493	0	39	0	245	274
494	0	39	0	246	275
495	0	39	0	247	276
496	0	39	0	248	253
497	0	39	0	249	254
498	0	39	0	250	255
499	0	39	0	251	256
500	0	39	0	252	257
501	0	0	0	253	277
502	0	0	0	254	277
503	0	0	0	255	277
504	0	0	0	256	277
505	0	0	0	257	277
506	0	0	0	258	277
507	0	0	0	259	277
508	0	0	0	260	277
509	0	4	4	261	277
510	0	4	4	262	277
511	0	4	4	263	277
512	0	4	4	264	277
513	0	4	4	265	277
514	0	4	4	266	277
515	0	4	4	267	277
516	0	4	4	268	277
517	0	4	4	269	277
518	0	4	3	270	277
519	0	4	0	271	277
520	0	4	0	272	277
521	0	4	0	273	277
522	0	4	0	274	277
523	0	4	0	275	277
524	0	4	0	276	277
525	0	39	39	277	12
526	0	39	39	12	1

APPENDIX C PROGRAM LIST

```
10 REM LNDG -- NUMBER OF LANDING IN OPERATION
20 REM TRUCK -- NUMBER OF TRUCKS AVALIABLE
30 REM N -- NUMBER OF NODES IN NETWORK, N = 25*LNDG + 27
40 REM M -- NUMBER OF ARCS IN NETWORK, M = 50*LNDG + 26
50 REM NODES NUMBERING TABLE
60 REM NODE(1) -- PARKING LOT
70 REM NODE(2) ... NODE(1+LNDG) -- LANDING NODES
80 REM NODE(2+LNDG) -- DUMMY NODE
90 REM NODE(3+LNDG) ... NODE(3+LNDG+24*LNDG-1) -- DEPARTURE
TIME NODES
100 REM NODE(25*LNDG+3) ... NODE(25*LNDG+3+23) -- ARRIVAL
TIME NODES
110 REM NODE(25*LNDG+27) -- TIMBER YARD NODE
120 REM TRUCK -- NUMBER OF TRUCKS
130 REM
140 REM
150 DIM X(600),C(600),L(600),U(600),P(600),A(600),B(600)
160 DIM I(600),J(600), LB(100), UB(100), T(100) ,NODE(600)
170 DIM
LANDING(600),FLOW1(600),FLOW2(600),TIME1(600),TIME2(600)
180 DIM
DEP(100),TRUCK1(100),TRUCK2(100),INV(100),LOADRATE(50)
190 PRINT "INPUT NUMBER OF TRUCK TYPES"
200 LPRINT "INPUT NUMBER OF TRUCK TYPES"
210 INPUT TYPE
220 PRINT "YOUR INPUT OF NUMBER OF TRUCK TYPES="TYPE,
"CORRECT?(Y/N)"
230 LPRINT "YOUR INPUT OF NUMBER OF TRUCK TYPES="TYPE,
"CORRECT?(Y/N)"
240 X$=INPUT$(1)
250 IF X$ <> "Y" THEN 190
260 PRINT
270 LPRINT
280 PRINT "INPUT NUMBER OF LANDINGS:"
290 LPRINT "INPUT NUMBER OF LANDINGS:"
300 INPUT LNDG
310 PRINT "YOUR INPUT OF NUMBER OF LANDINGS IS:" ; LNDG
320 LPRINT "YOUR INPUT OF NUMBER OF LANDINGS IS:" ; LNDG
330 PRINT
340 LPRINT
350 PRINT "IS THAT RIGHT ? (Y/N)"
360 LPRINT "IS THAT RIGHT ? (Y/N)"
370 PRINT
380 LPRINT
390 X$ = INPUT$(1)
400 IF X$ <> "Y" THEN 260
410 M = 50*LNDG + 26
420 N = 25*LNDG + 27
430 PRINT
```

```
440 REM ASSIGN EACH TIME NODE A VALUE WHICH IS THE
TIME,eg,10:00 O'CLOCK
450 FOR I=LNDG+3 TO 25*LNDG+26
460 NODE(I)=K MOD 24
470 'PRINT "NODE("I")="NODE(I)
480 K=K+1
490 NEXT I
500 PRINT "INPUT LOW BOUND, UPPER BOUND, COST, TRAVELLING
TIME:"
510 LPRINT "INPUT LOW BOUND, UPPER BOUND, COST, TRAVELLING
TIME:"
520 PRINT
530 LPRINT
540 FOR II = 1 TO LNDG
550 PRINT "LANDING"II":  LOW BOUND, UPPER BOUND, COST,
TRAVELLING TIME,LOADRATE "
560 LPRINT "LANDING" II":  LOW BOUND, UPPER BOUND, COST,
TRAVELLING TIME,LOADRATE
570 INPUT L(II), U(II), C(II), T(II),LOADRATE(II)
580 T(II) = INT(T(II)):LOADRATE(II)=INT(LOADRATE(II))
590 TRUCK=TRUCK+U(II)                           :REM TRUCK=NUMBER
OF TURNS
600 PRINT
610 LPRINT
620 PRINT "LOW BOUND, UPPER BOUND, COST, TRAVELLING TIME
,LOADRATE:"
630 LPRINT "LOW BOUND, UPPER BOUND, COST, TRAVELLING TIME
,LOADRATE:"
640 PRINT
650 LPRINT
660 PRINT L(II), U(II), C(II), T(II),LOADRATE(II)
670 LPRINT L(II), U(II), C(II), T(II),LOADRATE(II)
680 PRINT
690 LPRINT
700 PRINT "IS THAT RIGHT ? (Y/N)"
710 LPRINT "IS THAT RIGHT ? (Y/N)"
720 PRINT
730 LPRINT
740 X$ = INPUT$(1)
750 IF X$ <> "Y" THEN 550
760 NEXT II
770 PRINT "ENTER REVENUE PER TRUCK"
780 LPRINT "ENTER REVENUE PER TRUCK"
790 PRINT
800 LPRINT
810 INPUT REVERNUE
820 LPRINT "REVENUE="REVERNUE
830 REVERNUE=-REVERNUE
840 PRINT "INPUT REQUIREMENT OF YOURS FOR TRUCK ARRIVAL
DISTRIBUTION:"
850 LPRINT "INPUT REQUIREMENT OF YOURS FOR TRUCK ARRIVAL
DISTRIBUTION:"
860 PRINT
870 LPRINT
```

```
880 PRINT "  1. ALL THE LOW BOUND AND UPPER BOUND HAVE SAME
VALUES"
890 LPRINT "  1. ALL THE LOW BOUND AND UPPER BOUND HAVE SAME
VALUES"
900 PRINT "  2. EVERY LOW BOUND OR UPPER BOUND HAS DIFFERENT
VALUES"
910 LPRINT "  2. EVERY LOW BOUND OR UPPER BOUND HAS
DIFFERENT VALUES"
920 PRINT "       INPUT YOUR OPTION (1 / 2)"
930 LPRINT "       INPUT YOUR OPTION (1 / 2)"
940 PRINT
950 LPRINT
960 INPUT XX
970 IF XX = 1 THEN 990
980 IF XX = 2 THEN 1170 ELSE 840
990 PRINT
1000 LPRINT
1010 PRINT " INPUT LOW BOUND, UPPER BOUND VALUES"
1020 LPRINT " INPUT LOW BOUND, UPPER BOUND VALUES"
1030 INPUT LBB,UBB
1040 PRINT
1050 LPRINT
1060 PRINT LBB, UBB
1070 LPRINT LBB, UBB
1080 PRINT "IS THAT RIGHT ? (Y/N)"
1090 LPRINT "IS THAT RIGHT ? (Y/N)"
1100 X$ = INPUT$(1)
1110 IF X$ <> "Y" THEN 990
1120 FOR II = 1 TO 24
1130 LB(II) = LBB
1140 UB(II) = UBB
1150 NEXT II
1160 GOTO 1430
1170 FOR II = 1 TO 24
1180 PRINT
1190 REM
1200 LPRINT
1210 PRINT "INPUT REQUIREMENT FOR THE ";II;"-TH HOURS"
1220 LPRINT "INPUT REQUIREMENT FOR THE ";II;"-TH HOURS"
1230 INPUT LB(II), UB(II)
1240 PRINT
1250 LPRINT
1260 PRINT "THE   REQUIREMENT FOR THE ";II;"-TH HOURS ARE :"
1270 LPRINT "THE   REQUIREMENT FOR THE ";II;"-TH HOURS ARE
:"
1280 PRINT
1290 LPRINT
1300 PRINT LB(II), UB(II)
1310 LPRINT LB(II), UB(II)
1320 PRINT
1330 LPRINT
1340 PRINT "IS THAT RIGHT ? (Y/N)"
1350 LPRINT "IS THAT RIGHT ? (Y/N)"
1360 X$ = INPUT$(1)
```

```
1370 PRINT
1380 LPRINT
1390 IF X$ <> "Y" THEN 1210
1400 PRINT
1410 LPRINT
1420 NEXT II
1430 REM ARCS BETWEEN PARKING LOT AND   LANDINGS
1440 FOR II = 1 TO LNDG
1450 I(II) = 1
1460 J(II) = II+1
1470 NEXT II
1480 REM ARCS LINKING LANDINGS AND DUMP NODES
1490 FOR II = LNDG+1 TO 2*LNDG
1500 I(II) = II - LNDG + 1
1510 J(II) = 2 + LNDG
1520 L(II) = 0
1530 U(II) = TRUCK
1540 C(II) = 9999
1550 NEXT II
1560 REM ARCS BETWEEN LANDINGS    AND DEPARTURE TIME
1570 K1 = 2*LNDG
1580 K2 = 0
1590 FOR II = 1  TO LNDG
1600   FOR JJ = 1 TO 24
1610   K1 = K1 + 1
1620   K2 = K2 + 1
1630   I(K1) = II + 1
1640   J(K1) = 2 + LNDG + K2
1650   L(K1) = 0
1660   U(K1) = LOADRATE(II)
1670   C(K1) = 0
1680   NEXT JJ
1690 NEXT II
1700 REM ARCS  BETWEEN  DEPARTURE TIME AND ARRIVAL TIME
1710 K3 = 26*LNDG
1720 FOR II = 1 TO LNDG
1730   FOR JJ = 1 TO 24
1740   K3 = K3 + 1
1750   I(K3) = LNDG + 2 + JJ + (II-1)*24
1760   J(K3) = 25*LNDG + 3 + (T(II) + (JJ-1)) MOD 24
1770   L(K3) = 0
1780   U(K3) = TRUCK
1790   C(K3) = 0
1800   NEXT JJ
1810 NEXT II
1820 REM HOW MANY TIME INTERVALS NEEDED TO FINISH
TRASNPORTATION TASK
1830 USUM=0
1840 FOR I=1 TO LNDG
1850     IF C(I)+REVERNUE<0 THEN USUM=USUM+U(I) ELSE
USUM=USM+L(I)
1860 NEXT I
1870 REM (25*LNDG + 3) IS FIRST ARRIVAL NODE FROM LEFT
```

```
1880 REM ARCS BETWEEN ARRIVAL TIME NODES AND TIMBER YARD
NODE
1890 K4 = 50*LNDG
1900 Z=USUM/UBB                    :REM ADJUSTING ARRIVAL
TIME,SUITABLE
1910 FOR II = 1 TO 24               'FOR FIXING ARRIVAL
BOUNDARY ONLY
1920 K4 = K4 + 1
1930 I(K4) = 25*LNDG + 2 + II
1940 J(K4) = 25*LNDG + 27
1950 IF Z>=24 THEN 2010
1960 Z1=Z-16
1970 IF Z<16 THEN Z1=0
1980 IF II<=Z1 OR II>=9 THEN 2010
1990 L(K4)=0:U(K4)=0
2000 GOTO 2030
2010 L(K4) = LB(II)
2020 U(K4) = UB(II)
2030 C(K4) = REVERNUE
2040 NEXT II
2050 REM THE ARCS LINKING TIMBER YARD AND DUMMY NODE
2060 XX = 50*LNDG + 25
2070 I(XX) = 25*LNDG + 27
2080 J(XX) = LNDG + 2
2090 L(XX) = 0
2100 U(XX) = TRUCK
2110 C(XX) = 0
2120 REM THE ARC BETWEEN DUM NODE AND PARK LOT
2130 XXX=50*LNDG+26
2140 I(XXX)=LNDG+2:J(XXX)=1
2150 L(XXX)=0:U(XXX)=TRUCK:C(XXX)=0
2160 REM PRINT START NODE, FINISH NODE, LOW BOUND, UPPER
BOUND, COST
2170 'PRINT "ARC NUMBER  START NODE  END NODE LOW BOUND
UPBOUN COST"
2180 OPEN "B:example1.doc" FOR OUTPUT AS #1
2190 WRITE #1, M
2200 FOR II = 1 TO M
2210 WRITE #1, II,I(II), J(II), L(II), U(II), C(II)
2220 NEXT II
2230 RUNS=0
2240 RUNS=RUNS+1
2250 'FOR I=1 TO M
2260 'LPRINT USING "###      ###      ###      ###     ###
####.##";I,I(I),J(I),L(I),U(I),C(I)
2270 'NEXT I
2280 REM  -------------------------------------------------
------------
2290 REM  |              out of kilter algorithm
|
2300 REM  -------------------------------------------------
------------
2310 REM
2320 REM
```

```
2330 REM          ***************************
2340 REM          *      STEP  0             *
2350 REM          ***************************
2360 PRINT " OUT OF KILTER ALGORITHM "
2370 PRINT
2380 PRINT
2390 'PRINT "ENTER NUMBER OF NODES, NUMBER OF ARCS: " :
2400 'INPUT N, M
2410 'IF N>100 THEN 2100
2420 'IF N<1 THEN 2160
2430 'IF M>100 THEN 2130
2440 'IF M<1 THEN 2180
2450 'IF N>M THEN 2200
2460 'PRINT
2470 'PRINT "FOR EACH ARC, ENTER DATA IN THE FOLLOWING
ORDER: "
2480 'PRINT "START NODE, FINISH NODE, LOWER BOUND, UPPER
BOUND, COST"
2490 FOR M1=1 TO M
2500 X(M1) = 0
2510 'PRINT "ARC NUMBER"; M1;
2520 'INPUT I(M1), J(M1), L(M1), U(M1), C(M1)
2530 IF I(M1) > N THEN 2610
2540 IF I(M1) < 1 THEN 2630
2550 IF J(M1) > N THEN 2650
2560 IF J(M1) < 1 THEN 2670
2570 IF I(M1) = J(M1) THEN 2690
2580 IF L(M1) > U(M1) THEN 2710
2590 IF L(M1) < 0 THEN 2730
2600 GOTO 2750
2610 PRINT "START NODE NUMBER TOO LARGER, TRY AGAIN"
2620 GOTO 2510
2630 PRINT "START NODE NUMBER TOO SMALL, TRY AGAIN"
2640 GOTO 2510
2650 PRINT "FINISH NODE NUMBER TOO LARGER, TRY AGAIN"
2660 GOTO 2510
2670 PRINT "FINISH NODE NUMBER TOO SMALL, TRY AGAIN"
2680 GOTO 2510
2690 PRINT "START NODE AND FINISH NODE ARE IDENTICAL, TRY
AGAIN"
2700 GOTO 2510
2710 PRINT "LOWER BOUND EXCEEDS UPPER BOUND,TRY AGAIN"
2720 GOTO 2510
2730 PRINT "LOWER BOUND ON FLOW CANNOT BE NEGATIVE, TRY
AGAIN"
2740 GOTO 2510
2750 NEXT M1
2760 FOR N1 = 1 TO N
2770 P(N1) = 0
2780 NEXT N1
2790 REM   **************************
2800 REM   *      STEP  1           *
2810 REM   **************************
2820 FOR M1 = 1 TO M
```

```
2830   I1 = I(M1)
2840   J1 = J(M1)
2850   C1 = C(M1) + P(I1) - P(J1)
2860   IF C1 > 0 THEN  2930
2870   IF C1 < 0 THEN 2980
2880   B1 = U(M1) - X(M1)
2890   IF X(M1) < L(M1) THEN 3060
2900   B1 = X(M1) - L(M1)
2910   IF X(M1) > U(M1) THEN  3110
2920   GOTO 4200
2930   B1 = L(M1) - X(M1)
2940   IF X(M1) < L(M1) THEN 3060
2950   B1 = -B1
2960   IF X(M1) > L(M1) THEN 3110
2970   GOTO 4200
2980   B1 = U(M1) - X(M1)
2990   IF X(M1) < U(M1) THEN 3060
3000   B1 = -B1
3010   IF X(M1) > U(M1) THEN 3110
3020   GOTO 4200
3030   REM **************************
3040   REM *       STEP 2           *
3050   REM **************************
3060   S=J1
3070   T = I1
3080   A(S) = T
3090   B(S) = B1
3100   GOTO 3150
3110   S = I1
3120   T = J1
3130   A(S) = -T
3140   B(S) = B1
3150   FOR N1 = 1 TO N
3160     IF N1 = S THEN 3190
3170     A(N1) = 0
3180     B(N1) = 0
3190   NEXT N1
3200   M2 = 1
3210   M3 = M2
3220   FOR M4 = 1 TO M
3230   I4 = I(M4)
3240   J4 = J(M4)
3250   IF A(I4) = 0 THEN 3360
3260   IF A(J4) <> 0 THEN 3470
3270   IF X(M4) >= U(M4) THEN 3470
3280   B1 = U(M4) - X(M4)
3290   IF C(M4) + P(I4) - P(J4) <= 0 THEN  3320
3300   IF X(M4) >= L(M4) THEN 3470
3310   B1 = L(M4) - X(M4)
3320   A(J4) = I4
3330   IF B1>=B(I4) THEN B(J4)=B(I4) ELSE B(J4)=B1
3340  'B(J4) = MIN(B1, B1, B1, B(I4))
3350   GOTO 3450
3360   IF A(J4) = 0 THEN 3470
```

```
3370   IF X(M4) <= L(M4) THEN 3470
3380   B1 = X(M4) - L(M4)
3390   IF C(M4) + P(I4) - P(J4) >= 0 THEN 3420
3400   IF X(M4) <= U(M4) THEN 3470
3410   B1 = X(M4) - U(M4)
3420   A(I4) = -J4
3430  'B(I4) = MIN(B1, B1, B1, B(J4))
3440   IF B1>=B(J4) THEN B(I4)=B(J4) ELSE B(I4)=B1
3450   M2 = M2 + 1
3460   IF A(T) <> 0 THEN 3500
3470   NEXT M4
3480   IF M3 <> M2 THEN 3210
3490   GOTO 3700
3500   D = B(T)
3510 FOR I=1 TO N
3520 NEXT I
3530   N4 = A(T)
3540   K4 = ABS(N4)
3550 IF N4>0 THEN STARTNODE=K4:ENDNODE=T
3560 IF N4<0 THEN STARTNODE=T:ENDNODE=K4
3570 GOSUB 4510
3580   I4 = I(M4)
3590   J4 = J(M4)
3600   X(M4) = X(M4) + D*SGN(N4)
3610   N5 = I4                    :REM (previese node)
3620   IF N4 > 0 THEN 3640        :REM n4>0 means arc(i4,j4) is
forward arc
3630   N5 = J4                    :REM if arc(i4,j4) is reverse
arc
3640   IF N5 = T THEN 3690        :REM n5=t means D is added to
all arcs in chain
3650   N4 = A(N5)
3660 IF N4>0 THEN STARTNODE=ABS(N4):ENDNODE=ABS(N5)
3670 IF N4<0 THEN STARTNODE=ABS(N5):ENDNODE=ABS(N4)
3680   GOTO 3570
3690   GOTO 2830
3700   REM ******************************************
3710   REM *                STEP  3                 *
3720   REM ******************************************
3730   D = 9999
3740   FOR M4 = 1 TO M
3750   I4 = I(M4)
3760   J4 = J(M4)
3770   D1 = C(M4) + P(I4) - P(J4)
3780 IF D1=0 THEN 3910
3790 IF A(I4)=0 THEN 3860
3800 IF A(J4)<>0 THEN 3910
3810 IF D1<0 THEN 3910
3820 IF X(M4)>=U(M4) THEN 3910
3830 IF X(M4)<L(M4) THEN 3910
3840 IF D>D1 THEN D=D1
3850 GOTO 3910
3860 IF A(J4)=0 THEN 3910
3870 IF D1>0 THEN 3910
```

```
3880 IF X(M4)<=L(M4) THEN 3910
3890 IF X(M4)>U(M4) THEN 3910
3900 IF D>-D1 THEN D=-D1
3910   NEXT M4
3920   M5 = 0
3930   IF D < 9999 THEN 3990
3940   M5 = 1
3950   D = C(M1) + P(I1) - P(J1)
3960   D = ABS(D)
3970   IF X(M1) < L(M1) THEN 4340
3980   IF X(M1) > U(M1) THEN 4340
3990   FOR N1 = 1 TO N
4000     IF A(N1) <> 0 THEN 4020
4010     P(N1) = P(N1) + D
4020   NEXT N1
4030   IF M5 > 0 THEN 2830
4040   C1 = C(M1) + P(I1) - P(J1)
4050   IF A(S) > 0 THEN 4100
4060   B1 = X(M1) - L(M1)
4070   IF C1 >= 0 THEN 4130
4080   B1 = X(M1) - U(M1)
4090   GOTO 4130
4100   B1 = U(M1) - X(M1)
4110   IF C1 <= 0 THEN 4130
4120   B1 = L(M1) - X(M1)
4130   IF B1 = B(S) THEN 3210
4140   FOR N1 = 1 TO N
4150     IF A(N1) = 0 THEN 4180
4160     'B(N1) = MIN(B1, B1, B1, B(N1))
4170      IF B1<=B(N1) THEN B(N1)=B1
4180 NEXT N1
4190   GOTO 3210
4200   NEXT M1
4210   PRINT
4220   PRINT "                    OPTIMUM SOLUTION FOUND "
4230   PRINT
4240  ' LPRINT "LOWER BOUND     UPPER BOUND      FLOW
START NODE    END NODE  "
4250  FOR M1 = 1 TO M
4260  WRITE #1, L(M1), U(M1), X(M1),I(M1),J(M1)
4270  NEXT M1
4275 CLOSE #1
4280 PRINT
4290 'PRINT "NODE NUMBER     PI()    "
4300 'FOR N1 = 1 TO N
4310 '  PRINT N1, P(N1)
4320 'NEXT N1
4330 GOTO 4560
4340 LPRINT "THERE IS NO FEASIBLE SOLUTION "
4350 LPRINT "VALUES FOUND SO FAR"
4360 GOTO 4240
4370 GOTO 4560
4380 PRINT "NUMBER OF NODE CANNOT EXCEED 100"
```

```
4390 PRINT "PLEASE EITHER RE-ENTER, OR REDIMENSION THE
ARRAYS"
4400 GOTO 2380
4410 PRINT "NUMBER OF ARCS CANNOT EXCEED 100"
4420 PRINT "PLEASE EITHER RE-ENTER, OR REDIMENTION THE
ARRAYS"
4430 GOTO 2380
4440 PRINT "NUMBER OF NODES MUST BE POSITIVE, TRY AGAIN"
4450 GOTO 2380
4460 PRINT "NUMBER OF ARCS MUST BE POSITIVE, TRY AGAIN"
4470 GOTO 2380
4480 PRINT "NUMBER OF ARCS MUST EXCEED NUMBER OF NODES"
4490 PRINT "TRY AGAIN"
4500 GOTO 2380
4510 REM SUB SEARCHING ARCS IN FLOW AUGMENTING CHAIN
4520 FOR I=1 TO M
4530 IF I(I)=STARTNODE AND J(I)=ENDNODE THEN M4=I:GOTO  4550
4540 NEXT I
4550 RETURN
4560 REM PROGRAM FOR INTERPRETING RESULTS
4565 OPEN "B:RESULTS1.DOC" FOR OUTPUT AS #2
4570 lPRINT "
_____________________________________________________________

4580 lPRINT "                    DEPARTURE            |             |
ARRIVAL AT TIMBER YARD
4590 lPRINT "                    ______________________| TO LANDING
|_____________________________
4600 lPRINT "                 TIME |  # OF TRUCKS  |             |
TIME    |   # OF TRUCKS
4610 lPRINT "                    -------------------------------------
---------------------------
4620 K5=1
4630 FOR M1=1 TO M
4640 IF X(M1)=0 THEN 4710
4650 IF I(M1)<LNDG+3 OR I(M1)>25*LNDG+2 THEN 4710
4660 LANDING(K5)=INT((I(M1)-(LNDG+2)-
1)/24)+1:FLOW1(K5)=X(M1):FLOW2(K5)=X(M1)
4670
TIME1(K5)=NODE(I(M1)):TIME2(K5)=(NODE(I(M1))+T(LANDING(K5)))
MOD 24
4680 LPRINT USING "            ##          ##            ##
##          ##
";TIME1(K5),FLOW1(K5),LANDING(K5),TIME2(K5),FLOW2(K5)
4690 WRITE #2,
TIME1(K5),FLOW1(K5),LANDING(K5),TIME2(K5),FLOW2(K5)
4700 K5=K5+1
4710 NEXT M1
4720 lPRINT "
_____________________________________________________________

4730 lPRINT
4740 K5=K5-1            :REM DIDN'T CALCUL. TIME1(K5)...,BUT
1 WAS ADDED TO K5
```

```
4750                           REM AT LAST IN LOOP ABOVE
4760 REM SEPRATE THE NEXT DAY'S TRUCK DEPARTURE FROM TODAY'S
4770 COUNT=0
4780 FOR I=1 TO K5
4790      IF TIME2(I)>=TIME1(I) THEN 4840
4800      COUNT=COUNT+1
4810      SWAP TIME1(I),TIME1(K5-COUNT+1)
4820      SWAP FLOW1(I),FLOW1(K5-COUNT+1)
4830      SWAP LANDING(I),LANDING(K5-COUNT+1)
4840 IF I=K5-COUNT THEN 4860
4850 NEXT I
4860 REM SORT FOR TODAY'S TRUCK DEPARTURE
4870 FOR I=K5-COUNT+1 TO K5-1
4880      FOR J=I+1 TO K5
4890           IF TIME1(I)<TIME1(J) THEN 4930
4900           SWAP TIME1(I),TIME1(J)
4910           SWAP FLOW1(I),FLOW1(J)
4920           SWAP LANDING(I),LANDING(J)
4930      NEXT J
4940 NEXT I
4950 REM SORT FOR DEPARTURE TIME
4960 FOR I=1 TO K5-1-COUNT
4970      FOR J=I+1 TO K5-COUNT
4980           IF TIME1(I)<TIME1(J) THEN 5020
4990           SWAP TIME1(I),TIME1(J)
5000           SWAP FLOW1(I),FLOW1(J)
5010           SWAP LANDING(I),LANDING(J)
5020      NEXT J
5030 NEXT I
5040 REM SORT FOR ARRIVAL TIME
5050 FOR I=1 TO K5-1
5060      FOR J=I+1 TO K5
5070           IF TIME2(I)<TIME2(J) THEN 5100
5080           SWAP TIME2(I),TIME2(J)
5090           SWAP FLOW2(I),FLOW2(J)
5100      NEXT J
5110 NEXT I
5120 LPRINT
5130 LPRINT
5140 LPRINT "                                   RECOMMANDING LOG
TRUCK SCHEDULE "
5150 LPRINT "
____________________________________________________________________________
__________"
5160 LPRINT "        DEPARTURE TIME    # OF TRUCKS     LANDING
ARRIVAL TIME     # OF TRUCKS"
5170 LPRINT"          --------------    -----------     -------
------------     -----------"
5180 LPRINT
5190 FOR I=1 TO K5
5200 IF I<K5-COUNT+1 THEN 5230
5210 LPRINT "          **";
```

```
5220 LPRINT USING "    ###             ##             ##
###
##";TIME1(I),FLOW1(I),LANDING(I),TIME2(I),FLOW2(I):GOTO 5240
5230 LPRINT USING "                ###            ##
##         ###
##";TIME1(I),FLOW1(I),LANDING(I),TIME2(I),FLOW2(I)
5240 NEXT I
5250 LPRINT "
____________________________________________________________

__________
5252 LPRINT
5254 LPRINT "     * indicates that the departing time, # of"
5256 LPRINT " departure trucks and the landing sent to are "
5258 LPRINT " about the day before our planning day. "
5260 DIM FLW1(100),FLW2(100),TM1(100),TM2(100)
5270 I=1:II=1
5280 FLW1(II)=FLOW1(I):TM1(II)=TIME1(I)
5290 IF I=K5-COUNT THEN TIME=TM1(II) : GOTO 5380
5300 IF TIME1(I)<>TIME1(I+1) THEN 5350
5310 FLW1(II)=FLW1(II)+FLOW1(I+1)
5320 I=I+1
5330 IF I=K5-COUNT THEN TIME=TM1(I) : GOTO 5380
5340 GOTO 5300
5350 II=II+1:I=I+1
5360 GOTO 5280
5370 REM
5380 REM
5390 REM ADD 0'S
5400 FOR I=0 TO 23
5410     DEP(I)=I
5420 NEXT I
5430 II=1
5440 FOR I=0 TO 23
5450     TRUCK1(I)=0
5460     IF DEP(I)<>TM1(II) THEN 5490
5470     TRUCK1(I)=FLW1(II)
5480     II=II+1
5490 NEXT I
5500 REM
5510 I=1:II=1
5520 FLW2(II)=FLOW2(I):TM2(II)=TIME2(I)
5530 IF I=K5 THEN 5610
5540 IF TIME2(I)<>TIME2(I+1) THEN 5590
5550 FLW2(II)=FLW2(II)+FLOW2(I+1)
5560 I=I+1
5570 IF I=K5 THEN 5610
5580 GOTO 5540
5590 II=II+1:I=I+1
5600 GOTO 5520
5610 REM
5620 REM ADD 0'S
5630 FOR I=0 TO 23
5640     DEP(I)=I
5650 NEXT I
```

```
5660 II=1
5670 FOR I=0 TO 23
5680     TRUCK2(I)=0
5690     IF DEP(I)<>TM2(II) THEN 5720
5700     TRUCK2(I)=FLW2(II)
5710     II=II+1
5720 NEXT I
5730 REM
5740 FOR I=0 TO 23
5750 PRINT TRUCK1(I),TRUCK2(I)
5760 NEXT I
5770 REM
5780 FOR I=0 TO 23
5790     INV(I)=TRUCK1(I)-TRUCK2(I)
5800     IF INV(I)<0 THEN INV=INV+INV(I)
5810 NEXT I
5820 INV=-INV
5830 LPRINT
5840 LPRINT
5850 LPRINT "                          TRUCK INVENTORY ANALYSIS "
5860 LPRINT"
____________________________________________________________
____________"
5870 LPRINT"            TIME    NO. OF DEPARTURE TRUCKS
NO. OF ARRIVAL TRUCKS  DIFFERENCE
5880 LPRINT"          -----  ----------------------   ----
-----------------  ------------
5890 FOR I=0 TO 23
5900  LPRINT USING "             ##                ##
##                  ## ";I,TRUCK1(I),TRUCK2(I),INV(I)
5910 NEXT I
5920 'FOR I=K5-COUNT+1 TO K5
5930 '    LPRINT USING "            ##                ##
##                  ##";TIME2(I),0,FLOW2(I),-FLOW2(I)
5940 'NEXT I
5950 LPRINT"
____________________________________________________________
____________
5960 LPRINT "
TOTAL TRUCKS NEEDED="INV
5970 IF TYPE=1 THEN 6020
5980 IF RUNS=TYPE THEN 6020
5990 GOSUB 6030
6000 GOSUB 6120
6010 GOTO 2380:REM SHOULD GO TO 2380 IN BIG PROGRAM
6020 END
6030 REM SUB RESET BOUNDS FOR ARCS BETWEEM ARRIVAL TIME
NODES & TIMBER YARD NODE
6040 K4=50*LNDG   :REM K4=LAST ARC BETWEEN DEP AND ARRIVAL
TIME NODES
6050 FOR I=0 TO 23
6060 K4=K4+1
6070 IF U(K4)=0 THEN U(K4)=UBB
6080 U(K4)=U(K4)-TRUCK2(I)
```

```
6090 LPRINT "U("K4")="U(K4)
6100 NEXT I
6110 RETURN
6120 REM SUB RESET BOUNDS BETWEEN PARK LOT & LANDING NODES
6130 FOR I=1 TO LNDG
6140 L(I)=U(I)-X(I):U(I)=U(I)-X(I)
6150 LPRINT "L("I")="L(I),"U("I")="U(I)
6160 NEXT I
6170 RETURN
```