FTL REPORT R 89-4

PAPERS FROM THE MIT-INDUSTRY COOPERATIVE RESEARCH PROGRAM, 1989

Proceedings of the Annual Meeting
held at MIT

Zabat
Williamson
Lee
Kolb
Karlsson
Fujiwara
Belobaba

May 25-26, 1989

MIT

DEPARTMENT
OF
AERONAUTICS
&
ASTRONAUTIC

FLIGHT TRANSPORTATION
LABORATORY
Cambridge, Mass. 02139

TABLE OF CONTENTS

Integration of Overbooking Models into Yield Management

Grace Lyn B. Zabat

May 25, 1989

Outline of Presentation

- Basic Definitions
- Objectives
- Overbooking Models
- Simulation Results
- Conclusions

Overbooking

- accepting reservations in excess of capacity in order to minimize empty seats on flights for which demand existed

• offsets the effect of cancellations and no-shows between now and departure time

• trade off between the cost of denied boarding vs. the cost of spoilage

Denied Boarding

- **passenger who holds a confirmed reservation (with a valid ticket) but is unable to board the plane because the flight was oversold**

- **also refers to the event of a passenger being denied boarding**

Spoilage

- **empty seats on a flight which was closed out on the day of departure**

Cost of Denied Boarding

- cash or travel benefits (eg. free tickets or travel vouchers)

- hotels, meals, transportation costs (airport/hotel)

- goodwill

Cost of Spoilage

- revenue from the reserved seat which went out empty

Objective:

Maximize Net Revenue

= Passenger Revenue

- Denied Boarding Cost

Alternative Objectives:

Minimize Probability of Denied Boarding

Minimize Proportion of Denied Boardings to Total Boardings

Problem: Given no-show rates of each fare class which are statistically different, how does overbooking by fare class compare with overbooking by cabin capacity?

Overbooking by Cabin Capacity

- finds overbooking level or authorized capacity, AU, by dividing actual capacity by a function of the aggregate show-up rates of all fare classes

- allocates the reservation spaces (equal to AU) to the different fare classes by an optimization method for seat inventory control

Integrated Overbooking/ Seat Inventory Control

- finds seat allocations by fare class based on actual capacity

- overbooks the fare class allocations to find the fare class overbooking limits and then AU

- **looked at several options of overbooking by fare class**

- **compared these options with cabin overbooking using simulations**

- **used mean demand and standard deviation, mean show-up rate and fare of each fare class as inputs to simulations**

1. Overbooking by Cabin Capacity

Example: Cap = 100 seats

NS = 15%

OV = 18%

AU = 118 spaces

Use EMSR to find fare class limits

	Y	M	B	Q
x	35	23	40	46
s	12	7	12	14
F($)	320	290	250	170
BL	118	98	76	26
NP	20	22	50	-

BL = Booking Limit

NP = Nested Protection Level

2. EMSR Fare Class Overbooking

- incorporates OV_i into allocations by deflating relative revenues to account for different show-up rates

- calculates protection levels and adjusts for show-up differences at the same time

Example:

	Y	M	B	Q
OV_i	.77	.83	.90	.95
BL	112	100	80	27
NP	12	20	53	-

3. Modified EMSR Overbooking

- uses passenger show-up distribution instead of demand distribution in allocation process
- overbooks actual seat allocations by a function of the weighted average of the fare class show-up rates

Example:

	Y	M	B	Q
BL	121	101	78	28
NP	20	23	50	-

4. Overbooking by Partial Enumeration

- optimizes expected net revenue using non-linear programming formulation

Objective Function:

$$\text{Max } E[R] = \sum F_i X_i p_i \; - \; DBC \cdot E[DB]$$

$$\text{st.} \qquad \sum X_i p_i \leq CAP$$

where

R = net revenue
F_i = fare in each class i
X_i = number of seats in class i
p_i = mean show-up rate of class i
DBC = cost per denied boarding
DB = denied boarding

- **not monotonically increasing in each fare class --> may have multiple optimal solutions**

- **incorporates costs of denied boardings**

- **seat allocations are applicable to a distinct fare class structure instead of a nested structure**

- **distinct allocations are used as nesting variables**

- **optimization requires an initial solution**

Example:

	Y	M	B	Q
BL	116	85	65	20
NP	31	20	45	-

Simulation Scenario A:

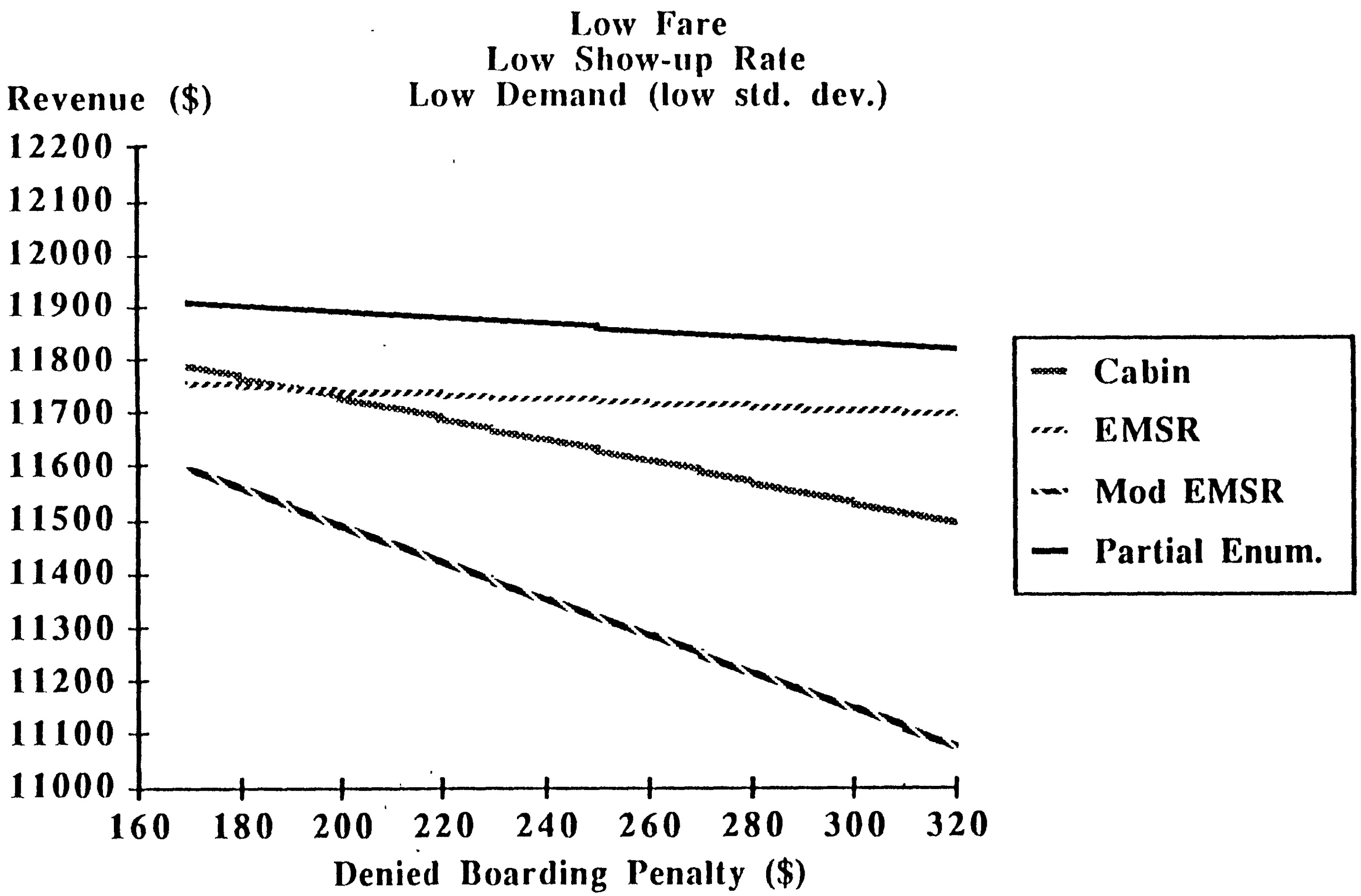
Low Demand

Low Standard Deviation

Low Show-up Rate

Low Fare

		Cabin	EMSR	Mod. EMSR	Partial Enum.
BL	Y	118	112	121	116
	M	98	100	101	85
	B	76	80	78	65
	Q	26	27	28	20
NP	Y	20	12	20	31
	M	22	20	23	20
	B	50	53	50	45
DB		1.97	0.37	3.43	0.60
SP		3.98	5.14	3.21	7.29


Low Fare
Low Show-up Rate
Low Demand (low std. dev.)
Revenue ($)
12200
12100
12000
11900
11800
11700
11600
11500
11400
11300
11200
11100
11000
160
180
200
220
240
260
280
300
320
Denied Boarding Penalty ($)
Cabin
EMSR
Mod EMSR
Partial Enum.

Simulation Scenario B:

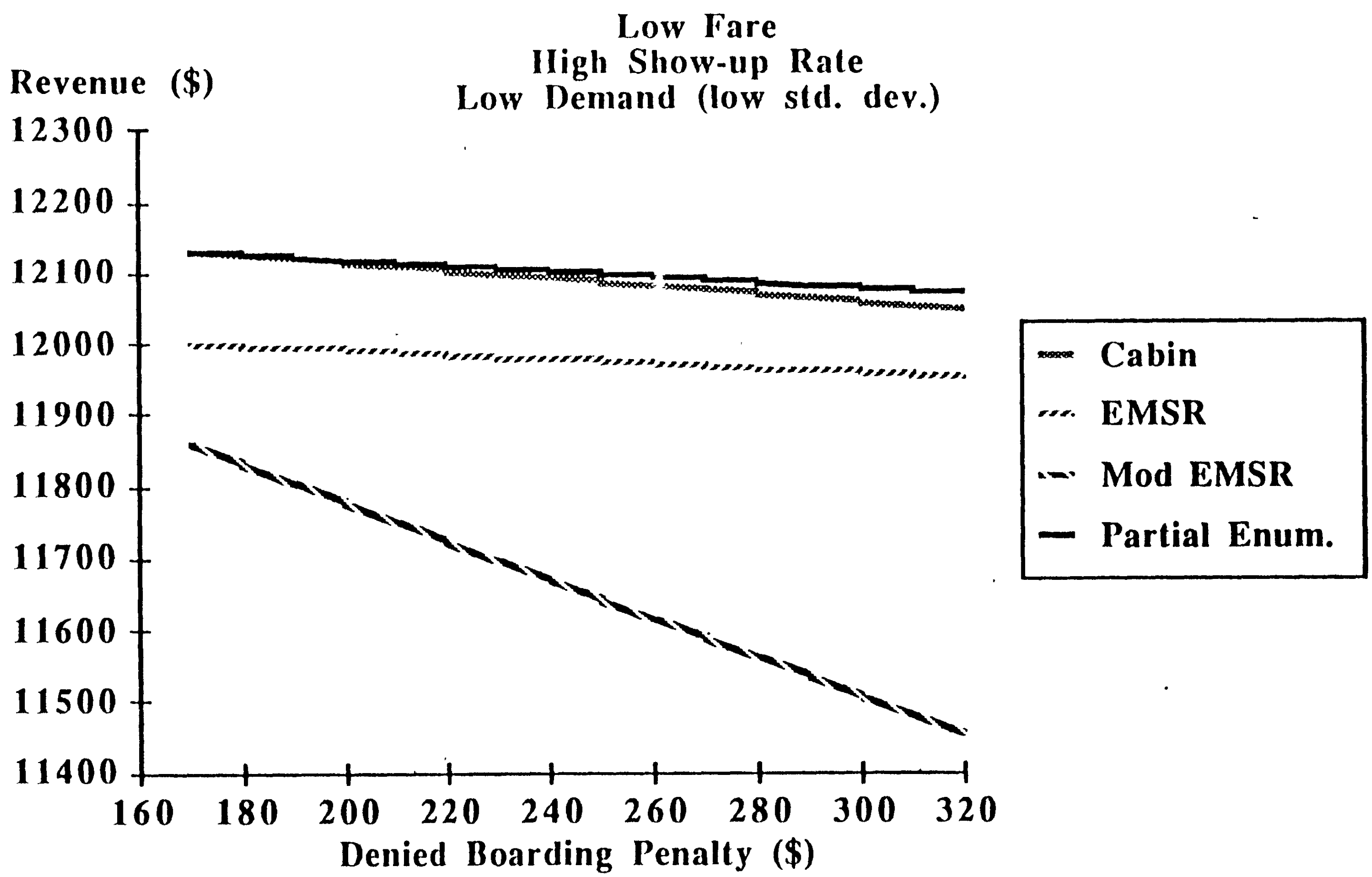
Low Demand

Low Standard Deviation

High Show-up Rate

Low Fare

		Cabin	EMSR	Mod. EMSR	Partial Enum.
BL	Y	112	110	117	113
	M	92	93	96	80
	B	70	71	73	59
	Q	20	23	24	14
NP	Y	20	17	21	33
	M	22	22	21	21
	B	50	48	49	45
DB		0.57	0.33	2.71	0.40
SP		5.53	5.47	3.73	8.69


Low Fare
High Show-up Rate
Low Demand (low std. dev.)
Revenue ($)
12300
12200
12100
12000
11900
11800
11700
11600
11500
11400
160
180
200
220
240
260
280
300
320
Denied Boarding Penalty ($)
Cabin
EMSR
Mod EMSR
Partial Enum.

Simulation Scenario C:

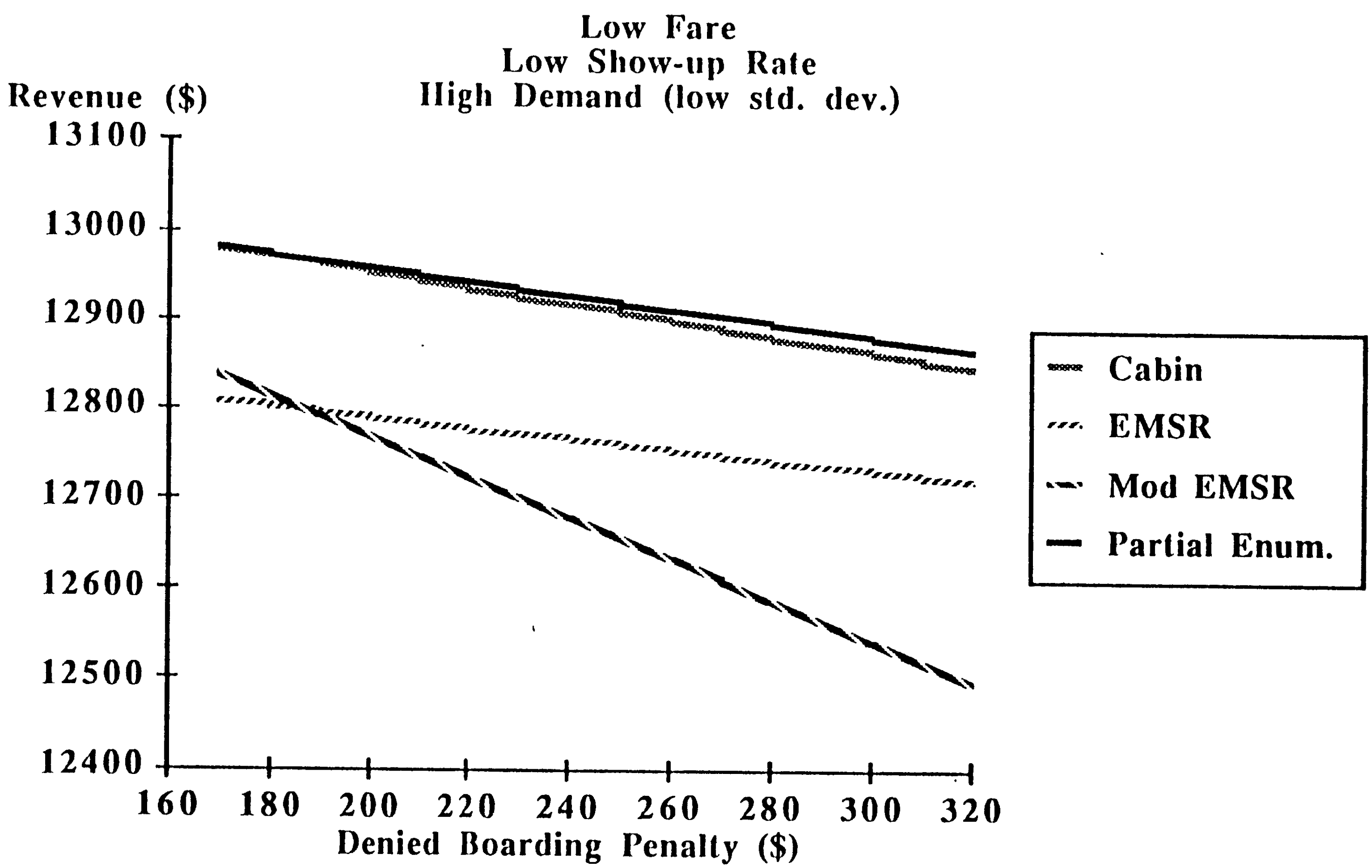
High Demand

Low Standard Deviation

Low Show-up Rate

Low Fare

		Cabin	EMSR	Mod. EMSR	Partial Enum.
BL	Y	118	115	122	118
	M	90	96	94	83
	B	62	71	65	59
	Q	0	4	2	0
NP	Y	28	19	28	35
	M	28	25	29	24
	B	62	67	63	59
DB		0.91	0.59	2.27	0.77
SP		5.99	5.62	4.89	6.51


Low Fare
Low Show-up Rate
High Demand (low std. dev.)
Revenue ($)
13100
13000
12900
12800
12700
12600
12500
12400
160
180
200
220
240
260
280
300
320
Denied Boarding Penalty ($)
Cabin
EMSR
Mod EMSR
Partial Enum.

Simulation Scenario D:

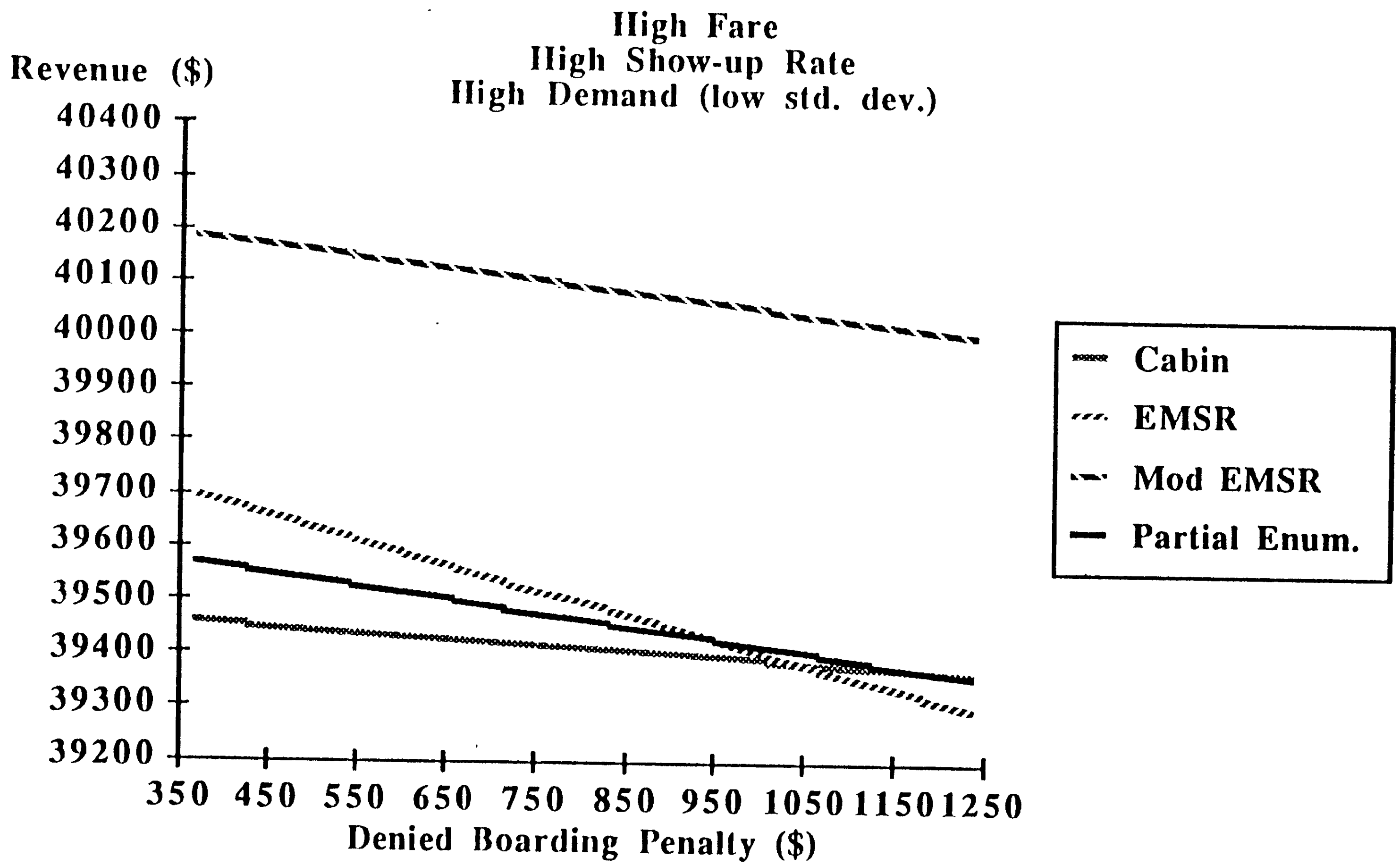
High Demand

Low Standard Deviation

High Show-up Rate

High Fare

		Cabin	EMSR	Mod. EMSR	Partial Enum.
BL	Y	112	115	113	116
	M	77	81	77	66
	B	29	33	29	21
	Q	0	3	0	0
NP	Y	35	34	36	50
	M	48	48	48	45
	B	29	30	29	21
DB		0.10	0.46	0.21	0.24
SP		13.58	10.35	12.10	17.77


High Fare
High Show-up Rate
High Demand (low std. dev.)
Revenue ($)
40400
40300
40200
40100
40000
39900
39800
39700
39600
39500
39400
39300
39200
350
450
550
650
750
850
950
1050
1150
1250
Denied Boarding Penalty ($)
Cabin
EMSR
Mod EMSR
Partial Enum.

Conclusions:

- Different methods have varying degrees of sensitivity to changes in input values

- Each method performs differently in each scenario

- Cabin overbooking gives reasonable results for changes in mean demand and show-up rates

- EMSR overbooking gives stable results but allocates more seats to lower classes

- **Modified EMSR overbooking gives best results for cases involving long-haul, high demand flights with high show-up rate for each fare class but results in higher denied boardings**

- **Partial enumeration gives best results for cases involving short-haul flights with low demand (with low std. dev.) and low show-up rates for each fare class and protects more seats for higher classes**

DEVELOPMENTS IN ORIGIN-DESTINATION SEAT INVENTORY CONTROL

Elizabeth L. Williamson

Flight Transportation Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Presented to
MIT-Industry Cooperative Research Program

May 25, 1989

PROBLEM:

To determine the number of seats to allocate to each origin-destination and fare class itinerary on each flight of an airline's route schedule in order to maximize total revenue.

Components of O-D Seat Inventory Control

OPTIMIZATION METHOD

CONTROL METHOD

O-D Approaches:

- Deterministic Network Optimization Method
- Probabilistic Network Optimization Method

DISTINCT DETERMINISTIC

- deterministic network formulation used to find seat allocation for each O-D/fare class over a network of flights

$$\text{Maximize} \sum_{O\text{-}D} \sum_{i} f_{i,O\text{-}D} \cdot x_{i,O\text{-}D}$$

Subject to:

$$\sum_{O\text{-}D} \sum_{i} x_{i,O\text{-}D} \leq CAP_j$$

for all O-D itineraries and *i* fare classes on flight *j*, for all flight *j*.

$$x_{i,O\text{-}D} \leq \mu_{i,O\text{-}D}$$

for all O-D itineraries and *i* fare classes.

- demand inputs reflect certainty, usually mean forecast demand for each O-D/fare class
- distinct booking limits applied to each O-D/fare class
- requires virtual classes for implementation

NESTED DETERMINISTIC

- use solution from distinct deterministic network formulation

- for control purposes, use distinct booking limits nest on the basis of:

  - Fare Classes
  - Fares
  - Shadow Prices

- booking limits by fare classes applied directly to reservations system booking classes while booking limits from other methods require virtual inventory classes

DISTINCT PROBABILISTIC

- probabilistic math programming formulation using (0,1) variables to represent each fare class/O-D/seat possibility in a <u>network</u> of flights

$$\text{Maximize} \sum_{O\text{-}D} \sum_{i} \sum_{j=1}^{CAP_k} EMR\left(j_{i,O\text{-}D}\right) \cdot x_{i,O\text{-}D,j}$$

Subject to:

$$\sum_{O\text{-}D} \sum_{i} \sum_{j=1}^{CAP_k} x_{i,O\text{-}D,j} \leq CAP_k$$

for all O-D itineraries and *i* classes on flight *k*, for all flights *k*

$$x_{i,O\text{-}D,j} = 0 \text{ or } 1$$

for all O-D itineraries, *i* classes, and $j=1,2,\ldots,CAP_k$

- forecast demand distributions used to generate expected marginal revenue for each variable.
- distinct booking limits for each O-D/fare class are produced
- requires virtual classes for implementation

NESTED PROBABILISTIC

- use solution from distinct probabilistic network formulation
- for control purposes, use distinct booking limits nest on the basis of:
  - Fare Classes
  - Fares
  - Deterministic Network Shadow Prices
- booking limits applied directly when nesting by fare class, otherwise virtual classes needed for implementation

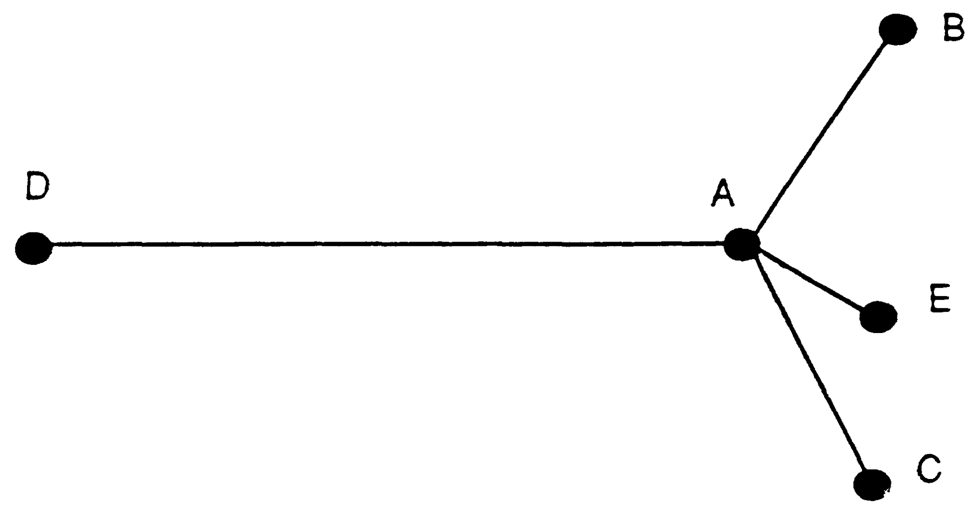


FIGURE 1: SMALL HUB NETWORK

TABLE 1: SIMULATION INPUT DATA - O-D CONTROL

CAPACITY = 150

MEDIUM DEMAND = 150 ON EACH LEG

O-D ITINERARIES		Y	M	B	Q
AB/BA	MEAN DEMAND	9	7	8	13
	STD. ERROR	2	2	3	3
	REVENUE	$310	$290	$95	$69
AE/EA	MEAN DEMAND	19	5	4	10
	STD. ERROR	3	2	1	3
	REVENUE	$159	$140	$64	$49
AC/CA	MEAN DEMAND	15	7	5	11
	STD. ERROR	3	3	2	2
	REVENUE	$280	$209	$94	$59
AD/DA	MEAN DEMAND	6	3	8	20
	STD. ERROR	2	2	2	5
	REVENUE	$455	$391	$142	$122
BE/EB	MEAN DEMAND	4	5	8	20
	STD. ERROR	2	2	4	6
	REVENUE	$319	$250	$109	$69
BC/CB	MEAN DEMAND	8	4	11	15
	STD. ERROR	3	2	3	3
	REVENUE	$403	$314	$124	$89
BD/DB	MEAN DEMAND	7	5	8	18
	STD. ERROR	2	3	3	4
	REVENUE	$575	$380	$159	$139
CE/EC	MEAN DEMAND	10	3	5	19
	STD. ERROR	4	2	2	5
	REVENUE	$226	$168	$84	$59
CD/DC	MEAN DEMAND	13	8	5	11
	STD. ERROR	3	3	1	2
	REVENUE	$477	$239	$139	$119
DE/ED	MEAN DEMAND	4	5	5	24
	STD. ERROR	2	3	2	7
	REVENUE	$502	$450	$154	$134

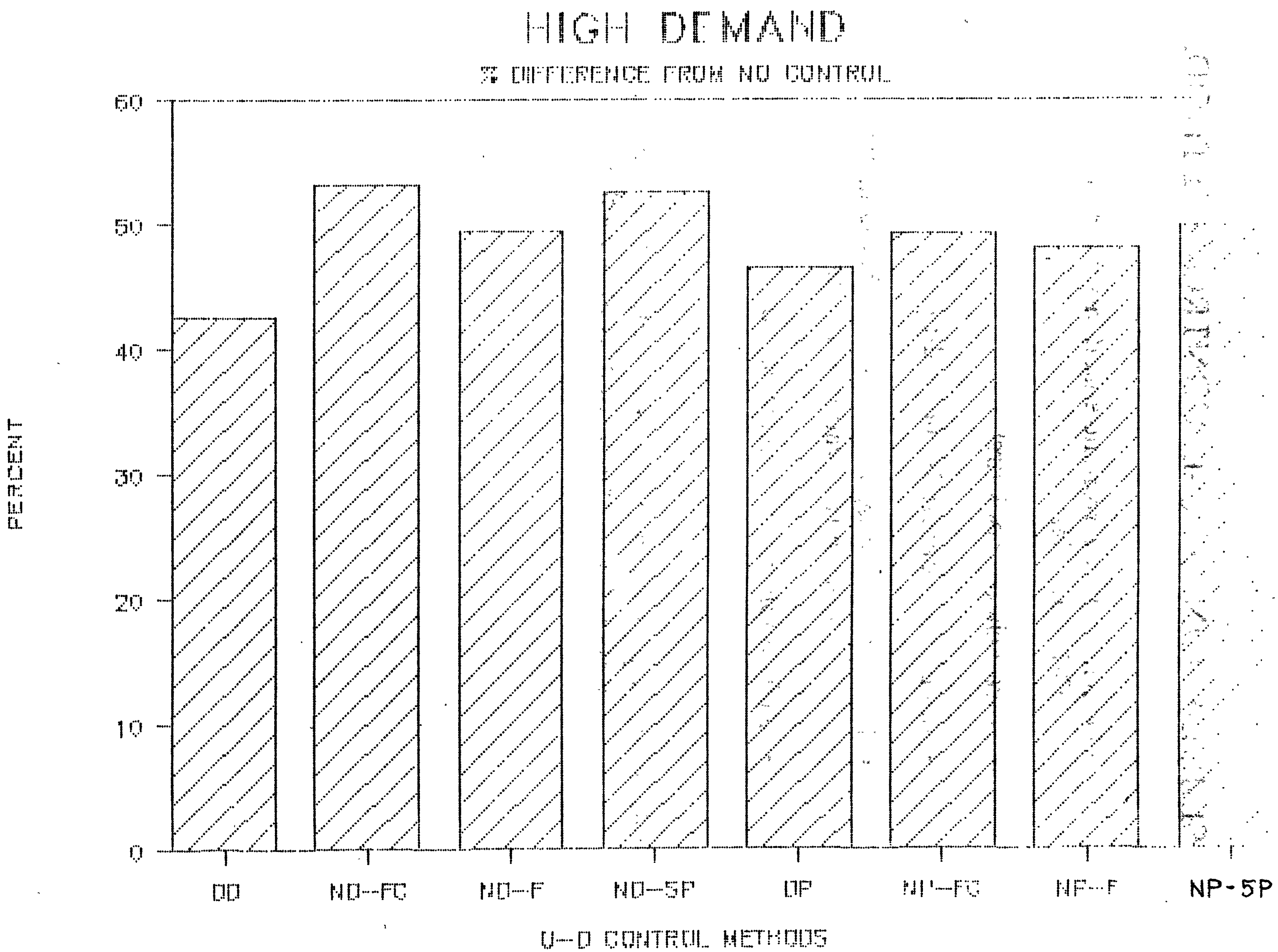
LOW DEMAND = 0.67 * MEDIUM

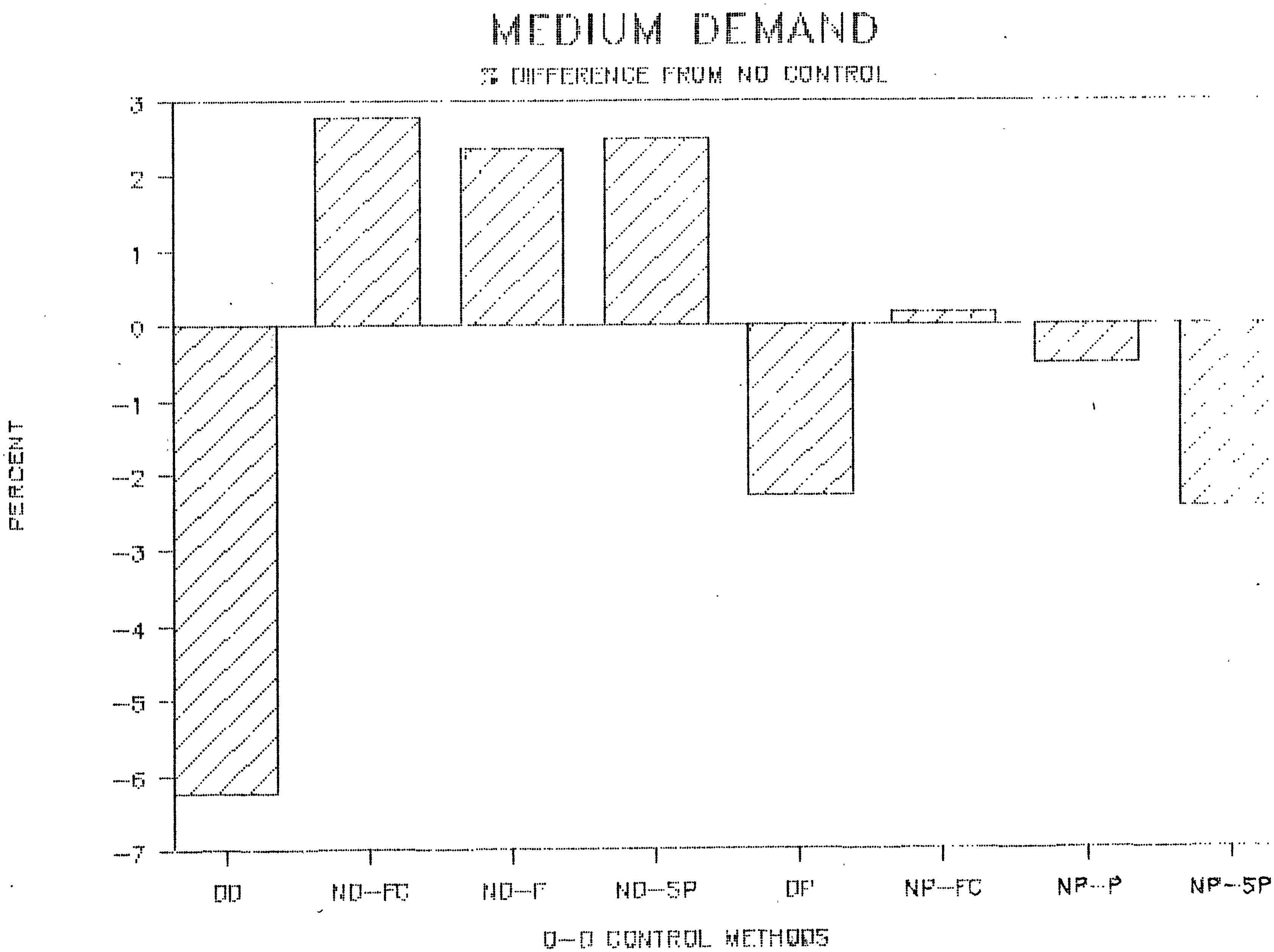
HIGH DEMAND = 1.33 * MEDIUM

SIMULATION OF O-D CONTROL METHODS

- Demands drawn from normal density for each O-D/fare class
- Lowest classes book first
- Independent demands; no "sell-up"
- Single point in time; no revisions of booking limits during reservations process
- 1,000 sample size (complete network)

OUTPUT: Expected total network revenue for each demand scenario and optimization/control method

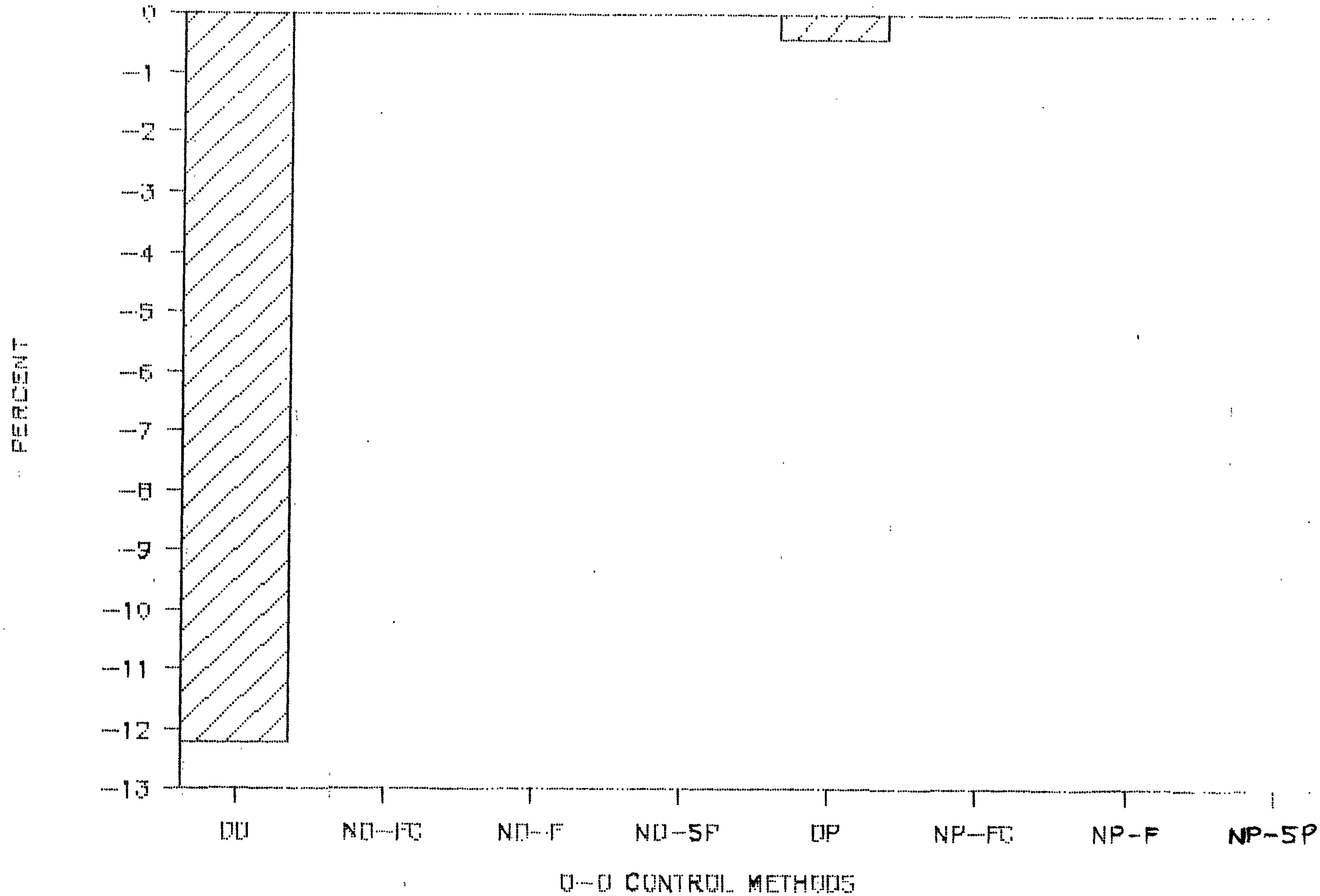

HIGH DEMAND
% DIFFERENCE FROM NO CONTROL
PERCENT
60
50
40
30
20
10
0
DD
ND-FC
ND-F
ND-5P
DP
NP-FC
NP-F
NP-5P
U-D CONTROL METHODS



MEDIUM DEMAND
% DIFFERENCE FROM NO CONTROL
PERCENT
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
OD
ND-FC
ND-F
ND-5P
OP
NP-FC
NP-P
NP-5P
O-D CONTROL METHODS

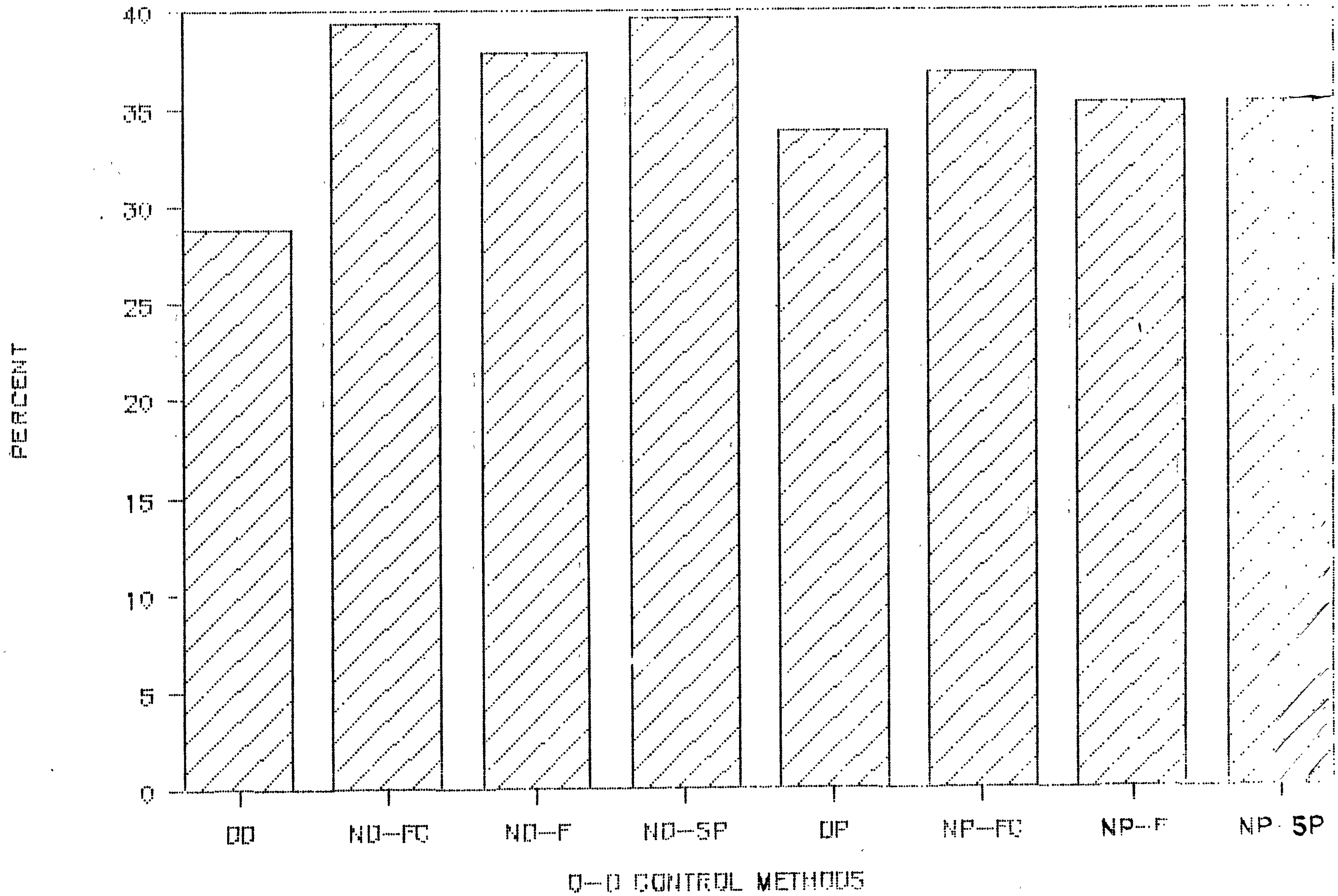
LOW DEMAND

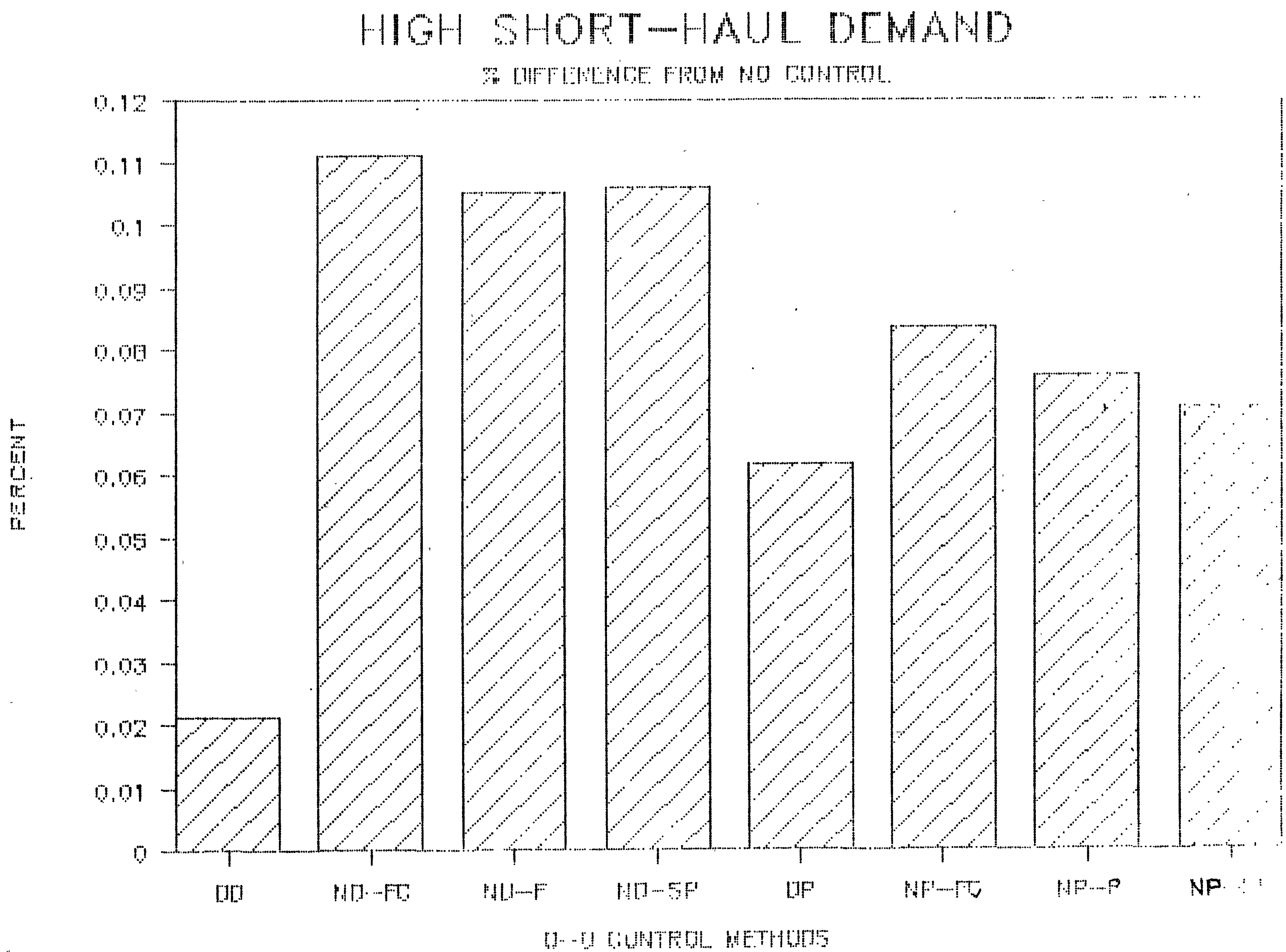
% DIFFERENCE FROM NO CONTROL



HIGH LONG-HAUL DEMAND

% DIFFERENCE FROM NO CONTROL




HIGH SHORT-HAUL DEMAND
% DIFFERENCE FROM NO CONTROL
0.12
0.11
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
PERCENT
DD
ND--FC
ND--F
ND--SP
DP
NP--FC
NP--P
O--D CONTROL METHODS

ORIGIN-DESTINATION CONTROL

- Combinations of optimization and control methods provide a wide variety of approaches to O-D seat inventory control.

- Important to understand how O-D optimization results can be implemented into the inventory control structure.

- Network formulations take into account the interaction of passenger flows across connecting flights in allocating seats.

- Poor matching of optimization and control methods can lead to negative revenue impacts.

- Given "small numbers" problems of O-D control, some form of aggregation and nesting become essential to improving revenues.

AIRLINE RESERVATIONS FORECASTING:

A PROGRESS REPORT

by

Anthony Lee

Flight Transportation Lab

and

Center for Transportation Studies

Massachusetts Institute of Technology

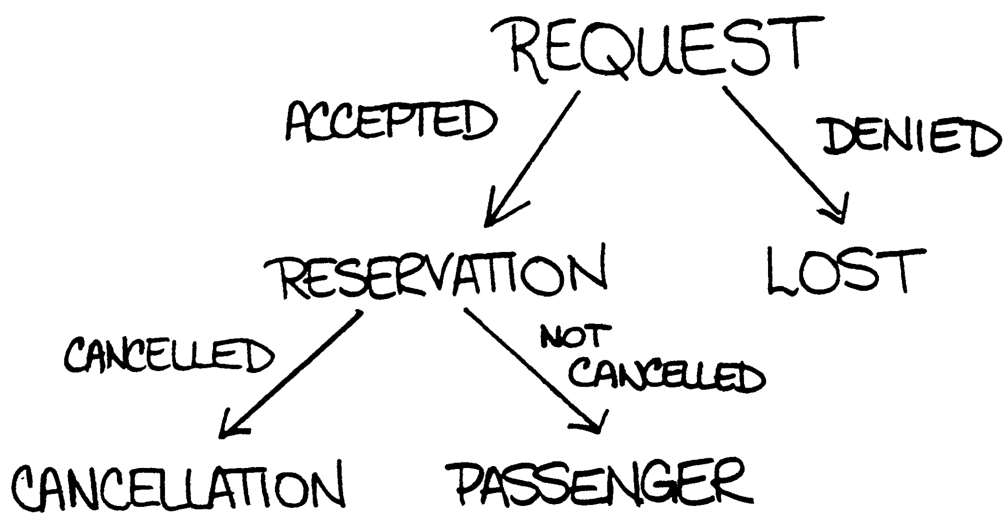
MIT - INDUSTRY COOPERATIVE RESEARCH PROGRAM

MAY 25, 1989

OUTLINE OF PRESENTATION

- BASIC DEFINITIONS
- GOAL AND MOTIVATION
- REVIEW OF MODELING FRAMEWORK
- COMPUTATIONAL RESULTS
- THE VALUE OF FORECASTING: SIMULATION RESULTS
- CONCLUSIONS

AIRLINE RESERVATIONS PROCESS



Bookings = Total Reservations - Total Cancellations

= number of reservations currently remaining in the system

$= B_{cfd}(t)$

<u>GOAL OF THIS PROJECT</u>

-- DEVELOP ACCURATE <u>FLIGHT-SPECIFIC</u>, <u>CLASS-SPECIFIC</u> FORECASTS OF FINAL BOOKINGS AT SPECIFIC TIMES BEFORE THE FLIGHT DEPARTS

-- EXAMPLE

TODAY, WE WANT TO FORECAST HOW MANY MORE FULL FARE (Y CLASS) PASSENGERS WILL BOOK ON FLIGHT 1234 DEPARTING ON JUNE 8.

$$= B_{Y,1234,6/8}(0) - B_{Y,1234,6/8}(14)$$

MOTIVATION

MORE PRECISE FORECASTS OF BOOKINGS

LEAD TO

BETTER INPUTS FOR REVENUE MAX. PROCEDURE

RESULTING IN

IMPROVED ALLOCATION OF SEATS AMONG FARE CLASSES

AND FINALLY

INCREASED REVENUES!!

HISTORICAL BOOKING MODEL

DATA: USES BOOKINGS ON PREVIOUS DEPARTURES OF THE SAME FLIGHT NUMBER

IDEA: CAPTURE TRENDS OVER TIME

MATHEMATICAL MODEL:

$$B_d(t) = \alpha_{0t} + \Sigma\, \alpha_{kt}\, B_{d-k}(t) + \Sigma\, a_{lt}\, \varepsilon_{d-l} + \varepsilon_d$$

EXAMPLE:

$$
\begin{aligned}
B_d(t) = \quad & b_1\text{*(time t bookings on date d-7)} + \\
& b_2\text{*(time t bookings on date d-14)} + \\
& b_3\text{*(time t bookings on date d-21)} + \\
& b_4\text{*(time t bookings on date d-28)} \\
& + \ldots + b_0
\end{aligned}
$$

ADVANCE BOOKING MODELS (BOOKING CURVES)

DATA: BOOKINGS ALREADY MADE ON A PARTICULAR FLIGHT AND OTHER FACTORS SUCH AS BOOKINGS IN LOWER CLASSES, SEASONALITY INDICES, ETC.

IDEA: CAPTURE RESERVATIONS BUILD UP OVER TIME

MODEL 1: (Pure Booking Curve Model)

$$B_d(t) \;=\; \beta_0 + f(t,\beta) + X\mathbf{b} + \eta_t$$

Bookings depend on time t before departure and other variables.

MODEL 2: (Time Series of Advance Bookings)

$$B_d(t) \;=\; \gamma_0 + \Sigma\, \gamma_l B_d(t+l) + X\mathbf{b} + \nu_t$$

Bookings depend on advance bookings in previous periods and other variables.

EXAMPLES:

- Pure Booking Curve Model

$$B_d(t) = a_1 * \log(t) + a_2 * (\text{seasonal index}) + a_3 * (\text{percentage sold in lower classes}) + \cdots + a_0$$

- Advance Bookings Model

$$B_d(0) = c_1 * (\text{bookings at time } t) + c_2 * (\text{bookings at time } t+7) + c_3 * (\text{bookings at time } t+14) + c_4 * (\text{bookings at time } t+21) + c_5 * (\text{seasonal index}) + c_6 * (\text{percentage sold in lower classes}) + \cdots + c_0$$

COMBINED MODEL:

IDEA: WEIGHTED COMBINATION OF HISTORICAL BOOKING MODEL AND ADVANCE BOOKING MODEL.

MODEL:

$$B_d(t) = \theta_1{}^* \text{ (historical booking model) } + \theta_2{}^* \text{ (advance booking model)}$$

$$= \alpha^*_{0t} + \Sigma\, \alpha^*_{kt} B_{d-k}(t) + \Sigma\, a^*_{lt}\, \varepsilon_{d-l} + \beta^*_0 + f(t,\beta^*) + \mathbf{X}b^* + \eta^*_t + \varepsilon^*_d$$

where $\alpha^* = \theta_1{}^* \alpha$, $\beta^* = \theta_2{}^* \beta$, and so forth.

EXAMPLE:

$$
\begin{aligned}
B_d(t) = \; & z_1 \text{*(time t bookings on date d-7)} + \\
& z_2 \text{*(time t bookings on date d-14)} + \\
& z_3 \text{*(time t bookings on date d-21)} + \\
& z_4 \text{*(time t bookings on date d-28)} + \\
& z_5 \text{*}\log(t) + \\
& z_6 \text{*(seasonal index)} + \\
& z_7 \text{*(percentage sold in lower} \\
& \qquad \text{classes)} + \cdots + z_0
\end{aligned}
$$

COMPUTATIONAL CONSIDERATIONS

DISTINGUISHING BETWEEN *ESTIMATION* AND *FORECASTING*

ESTIMATION fits a curve to past and current observations.

-- All observations are known values.

FORECASTING predicts future (unknown) values given the past and current observations.

-- Producing a forecast requires estimation and intelligent extrapolation.

CURRENT INDUSTRY PRACTICE IN FORECASTING

-- N-WEEK MOVING AVERAGE MODEL WITH SOME ADJUSTMENTS FOR SEASONALITY

-- MATHEMATICAL STATEMENT:

$$B_d(0) = B_d(t) + (1/8) \ \Sigma \ [B_{d-k}(0)-B_{d-k}(t)]$$

-- COMMENTS

1. ONE STEP ESTIMATION AND FORECASTING.
2. DOES NOT TAKE BOOKING CURVE EFFECT INTO ACCOUNT.
3. IT SEEMS THAT MORE RECENT OBSERVATIONS SHOULD BE MORE HEAVILY WEIGHTED.

COMPARISON OF MOVING AVERAGE MODELS AND REGRESSION MODELS

MOVING AVERAGE MODEL:

8 - WEEK AVERAGE OF BOOKINGS TO COME

REGRESSION MODELS:

HISTORICAL (TIME SERIES) MODEL

ADVANCE (BOOKING CURVE) MODEL

COMBINED MODEL

MEASURE OF ACCURACY OF FORECASTS:

MEAN SQUARE ERROR OF FORECAST =

$$(1/N) \ \Sigma(\text{FORECAST - ACTUAL})^2$$

SUMMARY OF RESULTS

TABLE 1: LAS-MSP MARKET (Q CLASS)

- ESTIMATION DATA: 6 MONTHS
- FORECASTING DATA: 4 MONTHS
- REGRESSION MODEL:

$$\begin{aligned} B_d(t) = \; & b_1\text{*(TIME)} + \\ & b_2\text{*(SEASONAL INDEX)} + \\ & b_3\text{*(BOOKINGS previous week)} + \\ & b_4 \end{aligned}$$

COMPARISON OF MEAN SQUARE ERROR OF FORECAST

MODEL	Day 7 to 0	Day 14 to 0	Day 21 to 0	Day 28 to 0
MA	10.52	15.14	20.01	23.04
REG	9.96	17.39	17.53	16.34
% IMPR.	5.23	-14.9	12.40	29.00

TABLE 2: BOS-MSP MARKET (M CLASS)

- ESTIMATION DATA: 6 MONTHS
- FORECASTING DATA: 4 MONTHS
- REGRESSION MODEL:

$$
\begin{aligned}
B_d(t) = \quad & b_1*(35\text{-TIME})^2 + \\
& b_2*(\text{SEASONAL INDEX}) + \\
& b_3*(\text{BOOKINGS previous week}) + \\
& b_4
\end{aligned}
$$

COMPARISON OF MEAN SQUARE ERROR OF FORECAST

MODEL	Day 7 to 0	Day 14 to 0	Day 21 to 0	Day 28 to 0
MA	5.20	10.80	12.85	14.61
REG	4.95	9.33	9.87	10.80
% IMPR.	4.81	13.61	23.19	26.08

TABLE 3: MKE-MSP MARKET (B CLASS)

- ESTIMATION DATA: 6 MONTHS
- FORECASTING DATA: 4 MONTHS
- REGRESSION MODEL:

$$
\begin{aligned}
B_d(t) = \quad & b_1*(35\text{-TIME})^2 + \\
& b_2*(\text{SEASONAL INDEX}) + \\
& b_3*(\text{BOOKINGS previous week}) + \\
& b_4
\end{aligned}
$$

COMPARISON OF MEAN SQUARE ERROR OF FORECAST

MODEL	Day 7 to 0	Day 14 to 0	Day 21 to 0	Day 28 to0
MA	4.00	4.25	5.95	4.90
REG	4.33	3.61	4.45	3.93
% IMPR.	-8.25	15.05	25.21	19.80

TABLE 4: DEN-MSP MARKET (Y CLASS)

- ESTIMATION DATA: 6 MONTHS
- FORECASTING DATA: 4 MONTHS
- REGRESSION MODEL:

$$B_d(t) = b_1*(35\text{-TIME})^2 + b_2*(\text{SEASONAL INDEX}) + b_3*(\text{BOOKINGS previous week}* \text{SEASONAL INDEX}) + b_4$$

COMPARISON OF MEAN SQUARE ERROR OF FORECAST

MODEL	Day 7 to 0	Day 14 to 0	Day 21 to 0	Day 28 to 0
MA	7.68	9.48	11.13	11.64
REG	6.40	7.08	9.08	10.67
% IMPR.	16.67	25.32	18.42	8.33

THE VALUE OF FORECASTING

Main Idea: Determine the impact of the forecast accuracy on the allocation of seats among the fare classes and, importantly, on expected revenues.

Question: How much in expected revenues do we gain with improved forecasts?

SIMULATION OF VALUE OF FORECASTING

- ACTUAL AIRLINE DATA (DEMANDS) WITH SOME MINOR ADJUSTMENTS

- FOUR FARE CLASSES Y, B, M, AND Q WITH REVENUES OF 100, 70, 50, AND 30.

- AIRCRAFT CAPACITIES OF 100, 200, & 300

- FOUR DEMAND SCENARIOS:
  - LOW (30% OF CAPACITY)
  - MEDIUM (60% OF CAPACITY)
  - HIGH (90% OF CAPACITY)
  - VERY HIGH (120% OF CAPACITY)

- SIMULATION OF 2000 FLIGHTS FOR EACH SCENARIO

SUMMARY OF THE SIMULATION

GOAL: TO TEST THE EFFECT ON EXPECTED REVENUES OF A FORECAST MEAN AND STANDARD DEVIATION WHICH DIFFER FROM THE ACTUAL MEAN AND STANDARD DEVIATION OF DEMAND

PROCEDURE:

STEP 1: USE THE *FORECAST* MEAN AND STD DEV TO CALULATE THE EMSR BOOKING LIMITS

FORECAST = FACTOR X ACTUAL

(WHERE FACTOR VARIES FROM 0.25 TO 5)

- CASE 1: VARY STD DEV ONLY
- CASE 2: VARY MEAN ONLY
- CASE 3: VARY MEAN AND STD DEV

STEP 2: DEMAND IS RANDOMLY GENERATED FROM A NORMAL DISTRIBUTION WITH THE *ACTUAL* MEAN AND STD. DEV. OF DEMAND

STEP 3: DEMAND IS REALIZED AT A SINGLE POINT IN TIME. Q PASSENGERS BOOK FIRST, THEN M CLASS, NEXT B CLASS, AND FINALLY Y CLASS BOOKS LAST.

STEP 4: WE REPEAT STEPS 2 AND 3 FOR 2000 FLIGHTS.

TABLE 1 - VALUE OF ACCURACY IN FORECASTING

FORECAST ACCURACY	PERCENTAGE CHANGE IN AVERAGE REVENUE FROM THE BASE CASE — DEMAND LEVELS		
	LOW	MEDIUM	HIGH
50% OF BASE CASE	-0.4%	-2%	-9%
75% OF BASE CASE	-0.3%	-1%	-3%
90% OF BASE CASE	0%	-0.3%	-1%
95% OF BASE CASE	0%	0%	0%
BASE CASE (PERFECT FORECAST)	---	---	---
105% OF BASE CASE	0%	0%	0%
110% OF BASE CASE	0%	-0.5%	-0.5%
125% OF BASE CASE	0%	-2%	-3%
150% OF BASE CASE	-0.5%	-9%	-10%
200% OF BASE CASE	-5%	-24%	-18%

CONCLUSIONS:

- ACCURATE FORECASTS ARE PARTICULARLY IMPORTANT ON HIGH AND VERY HIGH DEMAND FLIGHTS

- EXPECTED REVENUES ARE MORE SENSITIVE TO DIFFERENCES BETWEEN THE FORECAST AND ACTUAL MEAN THAN TO VARIATIONS IN THE ACTUAL AND FORECAST STD DEV

- EACH 10% OF FORECAST ACCURACY CAN BE WORTH 2% TO 4% IN EXPECTED REVENUES

- FORECASTS WHICH ARE WITHIN 10% OF THE ACTUAL VALUE ARE "GOOD ENOUGH".

RUBBER AIRPLANE

A Flexible Computer Aid for Conceptual Design Based on Constraint Propagation and Component Modeling

Mark A. Kolb

Massachusetts Institute of Technology

Cambridge, MA

Computer Applications in Engineering Design

Motivation:

- Speeds calculation procedure.
- Enhanced speed allows more iterations through the design path.
- Enable computer to manage complexity of design problem, freeing the engineer to develop "creative" solutions.

History:

- Computer-Aided Design (CAD) systems for geometric design. Best suited to detailed design.
- Sequential programs based on vehicle class. Single design path, implying limited range of applicability.
- Multiple sequential programs linked by a common database. Recently, expert systems have been applied to managing the interfaces between sub-programs. Still limited by sub-program directionality.

The Conceptual Design Task:

- Can be computationally intensive (particularly in aerospace engineering), thus prompting the use of computers.
- Demands flexibility, since the designer must develop solutions to unanticipated problems. It requires creativity and innovation.
- Is therefore unamenable to delineation, as in a sequential computer program. An alternate approach is needed.

A Question

How can the flexibility afforded by CAD systems for detailed design be acheived in computer tools for conceptual design?

One Answer

Advanced programming techniques, such as

- Object-Oriented Programming
- Constraint Propagation

Object-Oriented Programming

- **Based on descriptions of objects and their behavior.**
- **Objects are represented as *instances* of object *classes*, which serve as templates.**
- **Classes provide *instance variables*, which are state variables whose values are specific to each instance.**
- **Behavior is represented by *methods*, which are procedures specialized to a given class.**

Object-Oriented Programming: An Example

Class: *Automobile*

Instance Variables:

Color

Number-of-Doors

Engine-Size

Amount-of-Fuel

Number-of-Passengers

Methods:

Accelerate

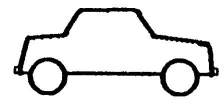
Decelerate

Add-Passenger

Remove-Passenger

Instances:

`car-1`

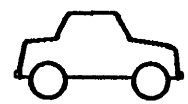


color: `red`

number-of-doors: `4`

engine-size: `V8`

`car-2`



color: `blue`

number-of-doors: `2`

engine-size: `V6`

Object-Oriented Programming: Inheritance

- **Definition of one class may be based upon the definition of another.**
- **A class inherits the instance variables and methods of its superclass.**
- **Inherited methods may be specialized to the new class.**

A collection of specialized methods for representing the same behavior over a range of classes serves to define a *generic operation.* **Generic operations provide a general-purpose interface for a given behavior which is context-sensitive: the particular response depends upon the class of the object to which it is applied.**

Inheritance: An Example

Superclass: *Moving-Object*

Provides instance variables for position and velocity.

Defines *Accelerate* **and** *Decelerate* **methods which change the object's velocity.**

Class: *Automobile*

Provides instance variables for color, engine size, etc.

Defines a specialized *Accelerate* **method which reduces the object's** *amount-of-fuel* **instance variable.**

Subclasses: *Off-Road Vehicle, Race Car, Passenger Vehicle*

Further specializations of the *Automobile* **class, which provide limits on engine size, the number of passengers, etc.**

Inheritance: An Example

Consider class *Cable Car*, a second subclass of *Moving Object*:

- **Cable cars carry no fuel.**
- **No need for a specialized *Accelerate* method.**

Consider class *Rocket*, a third subclass of *Moving Object*:

- **Rockets expend fuel when accelerating *and* decelerating.**
- **This class requires specialized methods for both the *Accelerate* operation and the *Decelerate* operation.**

Thus, the various *Accelerate* methods combine to form an *Accelerate* generic operation for *Moving Object* and its subclasses, whose behavior varies according to the type of moving object to which it is applied.

Applicability of Object-Oriented Programming

- **Design components (e.g., wings, fuselages, landing gear) are objects.**
- **Design involves the manipulation of these objects: positioning, sizing, etc.**
- **Design variables are also objects, with properties such as:**
  - **dimensionality**
  - **units**
  - **value**
  - **upper and lower bounds**
- **Mathematical relationships among these variables may be thought of as operating on their parameters, by manipulating parameter values in order to satisfy the relationship.**

Constraint Propagation

- **Conventional computer programs are comprised of sequences of *declarative* instructions:**

$$AR = b^2/S$$

- **Mathematically, such a statement also implies a number of equivalent *imperative* forms:**

$$b = \sqrt{AR \times S}$$

$$S = b^2/AR$$

- **A computer program which enforces one form must be re-written if an imperative form is needed.**

Constraint Propagation Systems:

- **Automatically infer the corresponding imperative forms from the given declarative input.**
- **Decide which form to apply, based on the available information.**
  Example: **Given** $AR = b^2/S$.
  If S **and** b **are known, then** $AR = b^2/S$.
  If AR **and** S **are known, then** $b = \sqrt{AR \times S}$.
  If AR **and** b **are known, then** $S = b^2/AR$.
- **Blur the distinction between "code" and "data".**
  Mathematical relationships are both data to be manipulated and code to be executed.

Applicability of Constraint Propagation

- Conceptual design depends upon satisfaction of mathematical relationships which model the problem.
- Due to the unpredictable nature of conceptual design, the required forms of these relationships will vary from problem to problem.
- Constraint propagation affords the flexibility which allows a single program to be applied to a variety of conceptual design problems.

Advantages of Constraint Propagation

- Allow focus on content rather than form.
- Greater accountability.

The Function-Modeling Approach: *Paper Airplane*

- **Direct application of constraint propagation.**
- **Problems described in terms of** *design variables* **and** *design functions* **between those design variables.**
- **User provides input values.**
  Program applies constraint propagation to use the design functions to compute values for as many of the remaining design variables as possible.

Shortcoming:

Unfortunately, this approach lacks sufficient organizational structure for the development of an efficient large-scale library of design knowledge.

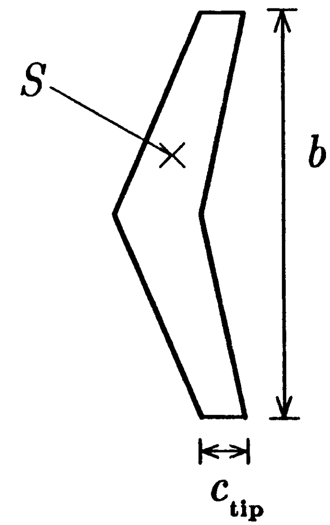
The Component-Modeling Approach: *Rubber Airplane*

- **A natural means for providing organizational structure in engineering design is the association of design functions and design variables with the *components* which they describe.**
- **The properties which describe components—dimensions, position, mass, performance characteristics, etc.—are represented as *attributes* of the components.**
- **The relationships which govern these properties are treated as *constraints* of the components.**
- **The design library may be built from component-classes, individual instances of which are created and manipulated by the designer.**
- **Permits the use of inheritance.**
- **Helps avoid name conflicts.**

Design Component Class: *Wing*

Attributes:

AR	aspect-ratio	c_{root}	root-chord
b	span	c_{tip}	tip-chord
S	wing-area	m	mass
Λ	sweep	L	lift
λ	taper-ratio	D	drag



Constraints:

$AR = b^2/S$	Definition of Aspect Ratio
$S = \frac{b}{2}(c_{root} + c_{tip})$	Calculation of Wing Area
$\lambda = c_{tip}/c_{root}$	Definition of Taper Ratio

Design Links

- **Need to account for constraints which relate the attributes of two or more components (e.g., attachment, relative sizing, and aerodynamic interference).**
- **Arbitrary assignment to one of the components is not modular. Introduce a new structure, the *design link*, which is defined similarly to components, in terms of its attributes and constraints.**
- **Constraints of design links may reference the attributes of components in addition to the attributes of the link.**

The relevant components are identified by specifying a set of *linkages* for the design link class.

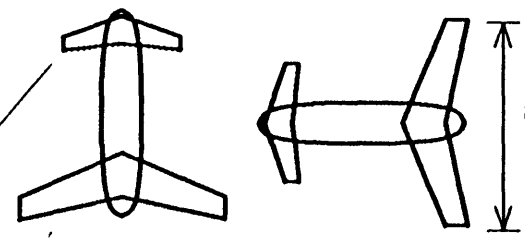
Design Links: An Example

An *Airfoil Interference* design link class would require a pair of linkages, one for the forward airfoil, and one for the rear airfoil.
Both linkages should be instances of the *Airfoil* class.

This *Airfoil Interference* design link might provide its own attribute for, say, the downwash on the two airfoils.
However, the constraints of this design link must also access various positional, dimensional, and aerodynamic attributes of the *Airfoil* instances themselves.

Design Link Class: *Airfoil Interference*

Instances:

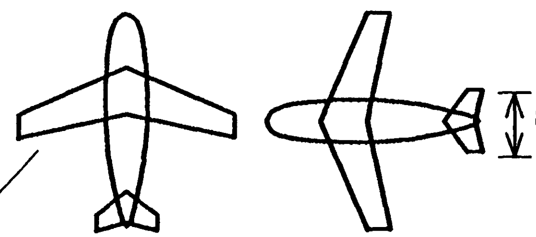


A. `Airfoil-Interference 1`

Linkages:

Forward-Airfoil → `Canard 1`

Rear-Airfoil → `Main-Wing 1`



B. `Airfoil-Interference 2`

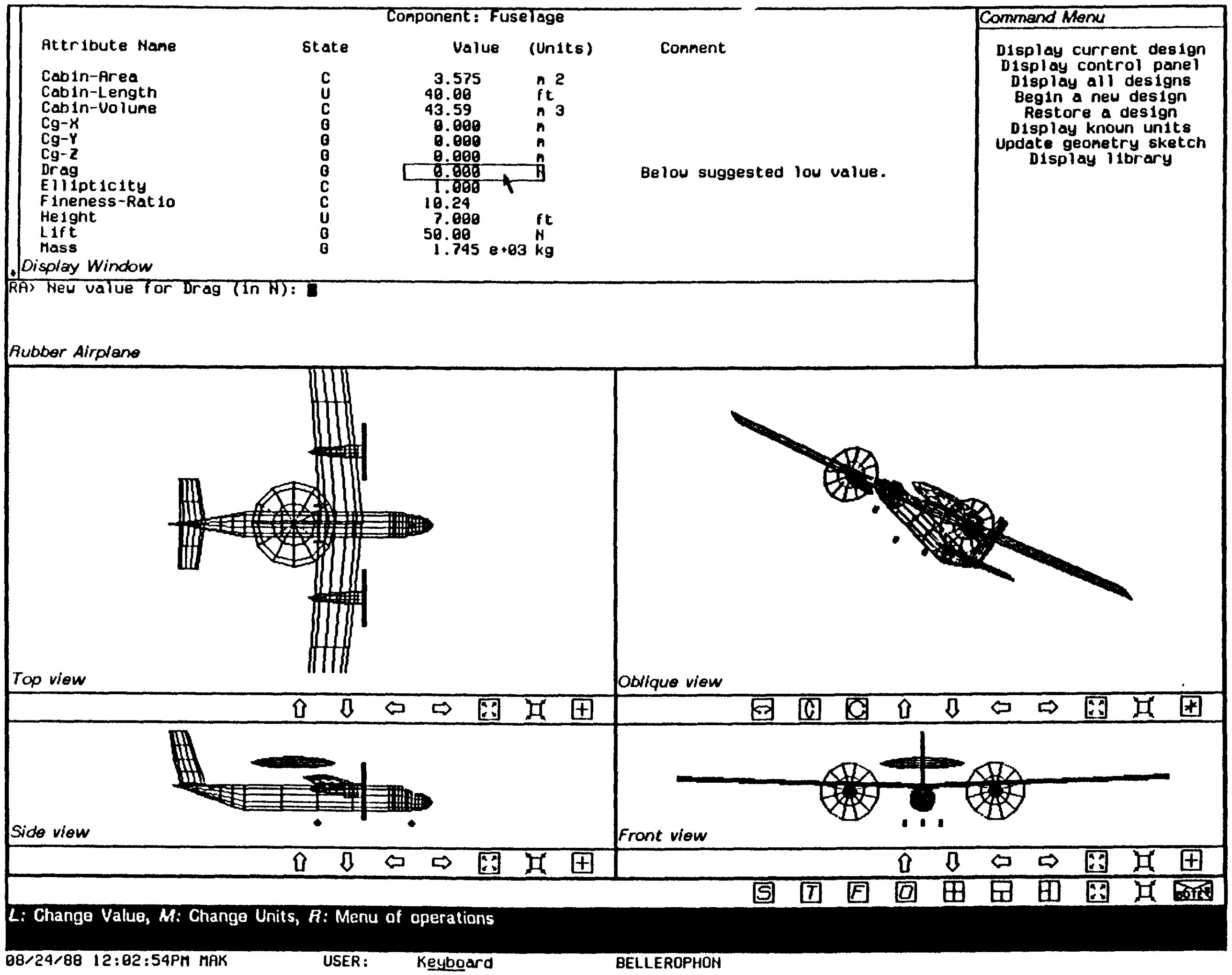
Linkages:

Forward-Airfoil → `Main-Wing 2`

Rear-Airfoil → `Vertical-Stabilizer 1`

Implementation Features

- Written in LISP.
- Components and links implemented as objects.
- Library comprised of component- and link-classes.
- Three-dimensional geometry display.
- Mouse-driven screen interface.
- Classes defined with text editor.

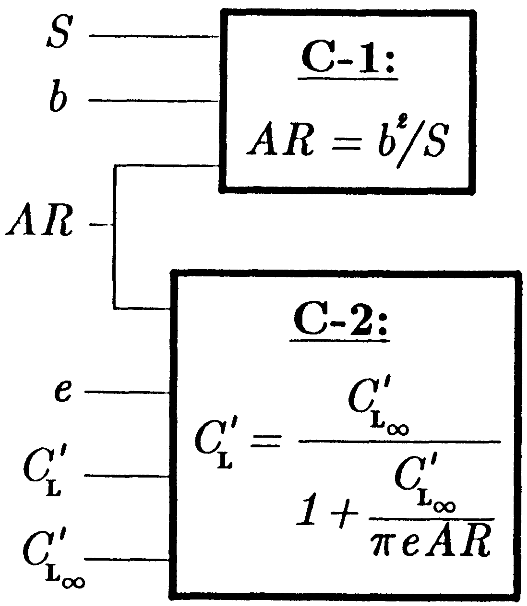

Component: Fuselage
Attribute Name
State
Value
(Units)
Comment
Cabin-Area C 3.575 m 2
Cabin-Length U 40.00 ft
Cabin-Volume C 43.59 m 3
Cg-X G 0.000 m
Cg-Y G 0.000 m
Cg-Z G 0.000 m
Drag G 0.000 N
Below suggested low value.
Ellipticity C 1.000
Fineness-Ratio C 10.24
Height U 7.000 ft
Lift G 50.00 N
Mass G 1.745 e+03 kg
Display Window
RA> New value for Drag (in N):
Rubber Airplane
Command Menu
Display current design
Display control panel
Display all designs
Begin a new design
Restore a design
Display known units
Update geometry sketch
Display library
Top view
Oblique view
Side view
Front view
L: Change Value, M: Change Units, R: Menu of operations
08/24/88 12:02:54PM MAK
USER:
Keyboard
BELLEROPHON

MAK

Constraint Propagation: Implementation

- **Each attribute is assigned a *value-supplier*:**
  - **a constraint**
  - **the symbol, ":user"**
  - **the symbol, ":guess"**
- **Initially, all value-suppliers are :guess.**
- **When the user provides a value for an attribute, its value-supplier becomes :user.**
- **Constraints for which exactly one parameter has a value-supplier of :guess are said to be *perfectly constrained*.**
- **Perfectly constrained constraints are used to compute the single free parameter, whose value-supplier becomes the constraint.**
- **This assignment may cause other constraints to become perfectly constrained, causing this procedure to be invoked recursively.**

Constraint Propagation: An Example



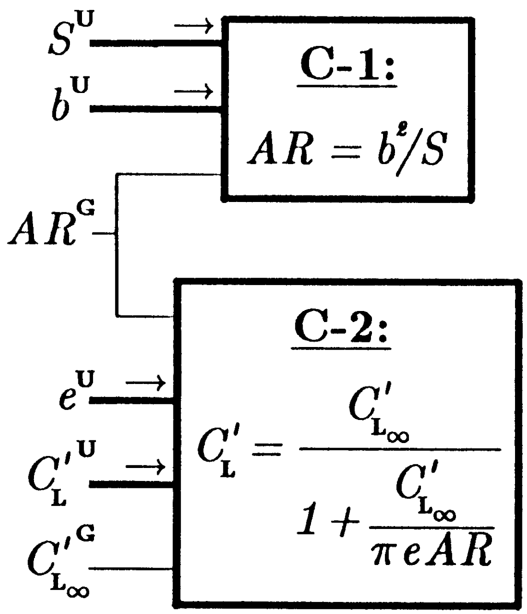
C-1: Definition of Aspect Ratio

Three parameters

C-2: Effect of finite span

Four parameters

Constraint Propagation: An Example



Initialization:

Planform Selected –

S, b, e **chosen by user**

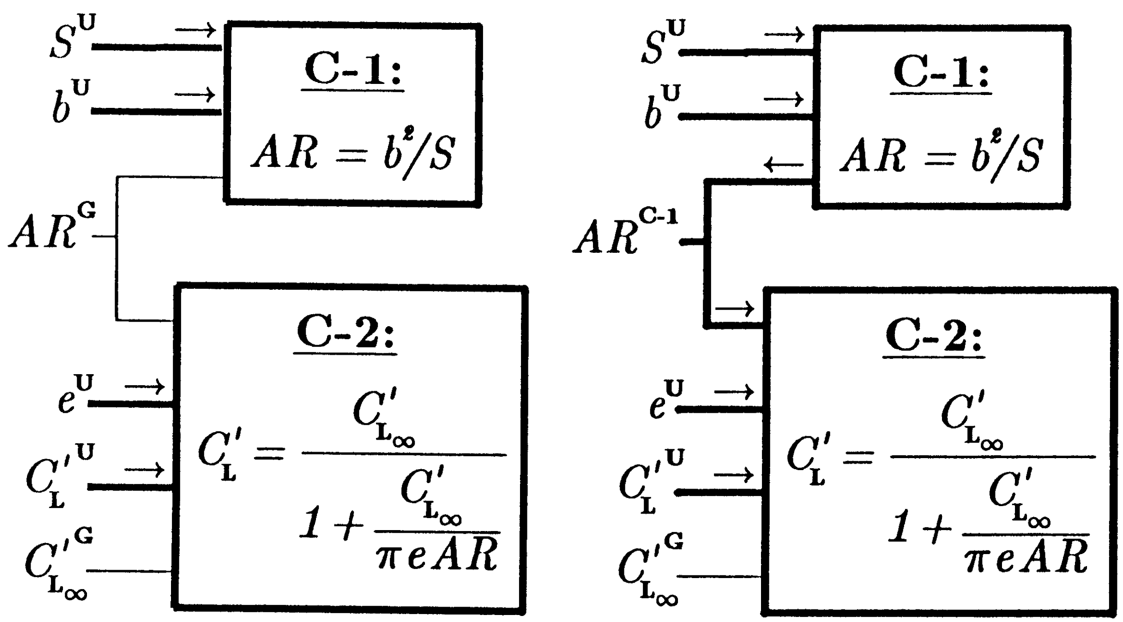
Required Performance Known –

C_L' **input by user**

Airfoil cross-section has not been selected.

AR **and** C_{L_∞}' **are initially unknown.**

Constraint Propagation: An Example



(a) (b)

(a)

Constraint C-1 is perfectly constrained.

(b)

Constraint C-1 has been applied to calculate *AR*.

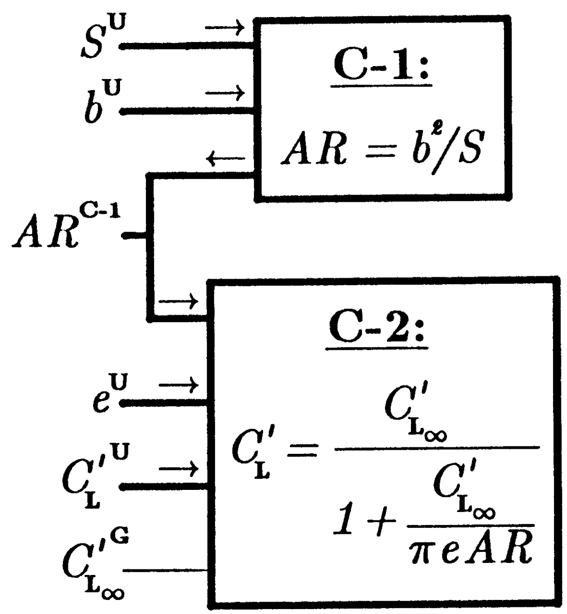
Constraint Propagation: An Example

(b)

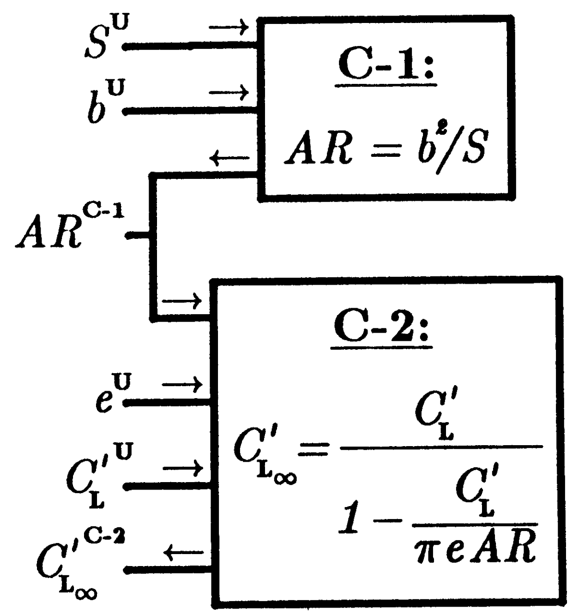
Constraint C-2 is perfectly constrained.

(c)

Constraint C-2 has been inverted and applied to calculate C'_{L_∞}.

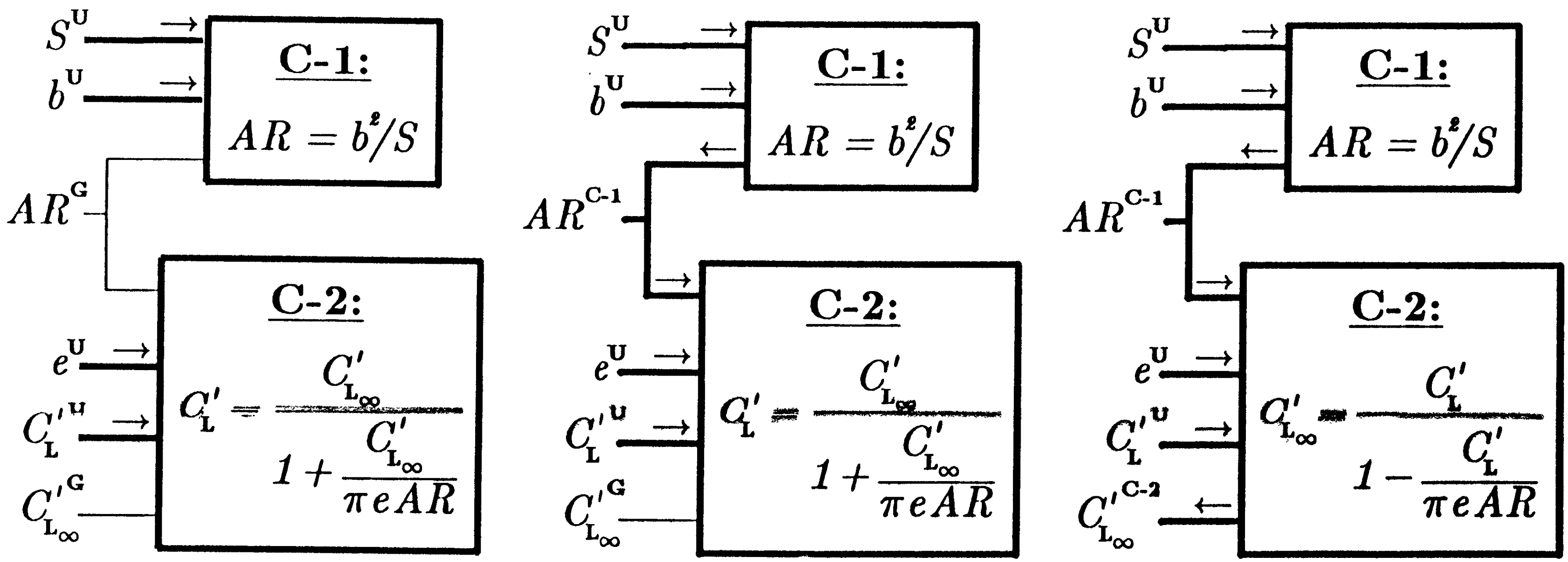


(b)



(c)

Constraint Propagation: An Example



(a) (b) (c)

Constraint Inversion: Implementation

- **Inversion rules based on:**
  - **individual primitive operations**
  - **recognized complex expressions**
  - **combination**
- **Numerical simulation**

Constraint Inversion: Numerical Implementation

- **Constraints are defined declaratively:**

$$y = f(x_1, x_2, \ldots)$$

- **To compute y, apply the constraint as defined.**
- **To compute x_k, where $x_k \in \{x_1, x_2, \ldots\}$, solve**

$$0 = f(x_1, x_2, \ldots) - y$$

for x_k holding all other variables constant.

Simultaneous Equations: Implementation

- Based on the "GTOW heuristic": computational loops.
- Modify first-order constraint propagation to note the ":guess attributes" of constraints with two or more such free parameters.
- After first-order constraint propagation is completed, check these attributes for the existence of loops.
- Solve loops numerically.

Loop Detection: Implementation

- Given an attribute with constraints of two or more :guess attributes.
- Assume a value-supplier for this attribute.
- Apply first-order constraint propagation of assumed value-suppliers, beginning with the original attribute, until:
  - An unused constraint with zero free parameters is encountered.
    The loop has been closed.
  - No further first-order propagation is possible.
    No loop exists.

Simultaneous Equations: Revised Implementation

- Original implementation relied solely on Newton-Raphson iteration. Found to be unstable for larger sets of simultaneous equations.
- Expand upon the "GTOW heuristic": use simple iteration for large loops.
- Requires selection of suitable iteration variable, as well as appropriate ordering of the loop constraints.

Iteration Loops: Implementation

- Detect loops as before.
- Search loops for appropriate iteration variable.
- For stability, order constraints according to original degrees of freedom.

Iteration Loops: An Example

- Consider the following set of equations:

$$
\begin{aligned}
W_{\text{engine}} &= constant \\
W_{\text{wing}} &= W_{\text{wing}}(AR, b, \Lambda, W_{\text{dry}}) \\
W_{\text{tail}} &= W_{\text{tail}}(W_{\text{wing}}, W_{\text{dry}}) \\
W_{\text{fuselage}} &= W_{\text{fuselage}}(l_{\text{fuselage}}, W_{\text{dry}}) \\
W_{\text{gear}} &= W_{\text{gear}}(W_{\text{dry}}) \\
W_{\text{dry}} &= W_{\text{engine}} + W_{\text{wing}} + W_{\text{tail}} + W_{\text{fuselage}} + W_{\text{gear}}
\end{aligned}
$$

- Assume geometry is known.
  Re-writing the equations with the known variables eliminated,

$$
\begin{aligned}
W_{\text{wing}} &= W_{\text{wing}}(W_{\text{dry}}) \\
W_{\text{tail}} &= W_{\text{tail}}(W_{\text{wing}}, W_{\text{dry}}) \\
W_{\text{fuselage}} &= W_{\text{fuselage}}(W_{\text{dry}}) \\
W_{\text{gear}} &= W_{\text{gear}}(W_{\text{dry}}) \\
W_{\text{dry}} &= W_{\text{dry}}(W_{\text{wing}}, W_{\text{tail}}, W_{\text{fuselage}}, W_{\text{gear}})
\end{aligned}
$$

Iteration Loops: An Example (Continued)

- Order constraints according to the number of unknowns prior to loop processing:

$$
\begin{aligned}
W_{\text{wing}} &= W_{\text{wing}}(W_{\text{dry}}) \\
W_{\text{fuselage}} &= W_{\text{fuselage}}(W_{\text{dry}}) \\
W_{\text{gear}} &= W_{\text{gear}}(W_{\text{dry}}) \\
W_{\text{tail}} &= W_{\text{tail}}(W_{\text{wing}}, W_{\text{dry}}) \\
W_{\text{dry}} &= W_{\text{dry}}(W_{\text{wing}}, W_{\text{tail}}, W_{\text{fuselage}}, W_{\text{gear}})
\end{aligned}
$$

  This ordering enhances stability, thus promoting convergence.

- Most frequently occurring unknown is W_{dry}.
  Select W_{dry} as the iteration variable. A value will be assumed for W_{dry}, and this assumed value is propagated until a new value for W_{dry} can be computed.

- New values for W_{dry} are iteratively propagated until convergence is obtained.

MAK

Initial Results: Test Cases

Current Test Case:

- **Long-Endurance, Manned Surveillance Aircraft.**
  Focus to date on mission performance.
  Current efforts directed towards aerodynamic analysis.

Planned Test Cases:

- **General Aviation Aircraft**
- **Small-Payload Launch Vehicle**

Initial Results

Observation:

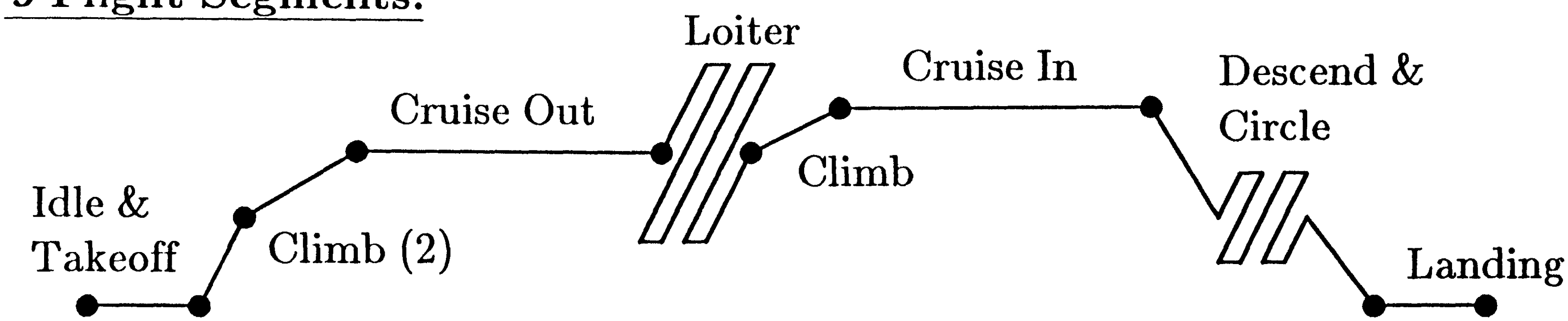
- **Flexibility is enhanced by limiting the number of attributes associated with components.**

Implications:

- **Component attributes are limited to geometry and gross properties.**
- **Secondary or derived properties relegated to design links.**
- **Special-purpose component properties are best treated as attributes of special-purpose design links.**

Design Link Class: *Loiter Mission*

9 Flight Segments:



Fuel calculations require C_L, L/D, SFC, etc., for each flight segment.

MAK

Initial Results

Observation:

- Component constraints can only access geometric and gross-property attributes.

Implications:

- Component constraints do not have sufficient information to compute gross properties.
- Component constraints are therefore primarily concerned with the calculation of geometric properties.
- Design links must be used to compute gross properties.

Initial Results

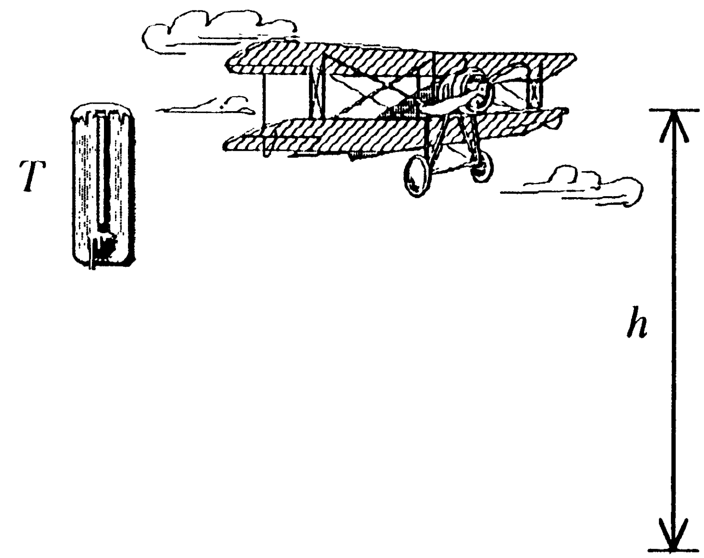
Observation:

- *Ad hoc* **single-mission analysis link is non-modular.**

Implications:

- **Better means for handling time-dependence is required.**
- **One approach: component-like "states" to account for time-varying attributes.**

Design State Class: *Atmosphere*



Attributes:

h	altitude
p	pressure
ρ	density
R	gas-constant
T	temperature

Constraints:

$\rho = \rho(h)$	Density of Standard Atmosphere
$p = p(h)$	Pressure of Standard Atmosphere
$T = T(h)$	Temperature of Standard Atmosphere
$p = \rho R T$	State Equation

MAK

Initial Results

Observation:

- Certain calculations are strongly coupled. The multiple-input, single-output paradigm is inadequate for representing such calculations.
- Some computations are difficult—if not impossible—to invert.

Implications:

- Need support for uni-directional constraints.
- Need support for multiple-input, multiple-output constraints.

Initial Results: Conclusion

Better means for handling complex constraints is required:

- "State" objects to account for time-dependent phenomena.
- Uni-directional and multiple-output constraints.

Flexibility is best served by:

- General-Purpose Component Classes—
  Component definitions tend to focus on geometry.
  Components also provide readily accessible attributes for important gross properties.
- Application-specific Link Classes—
  Design links are used for calculations such as performance analysis, weight determination, and relative locations.

Areas for Improvement:

- **Integration with a symbolic mathematics package.**
- **Improved interface for defining component- and link-classes.**
- **Enhanced graphics capabilities.**

Possibilities for Future Work:

- *Rubber Airplane* **as a platform for enhanced systems.**
- **Expert system for managing optimization, guiding performance function selection.**
- **Tool for translating design specifications into a set of components and links which serve as a baseline design.**

Automatic Speech Recognition in Air Traffic Control

Joakim Karlsson
Flight Transportation Laboratory
Massachusetts Institute of Technology

Automatic Speech Recognition (ASR) technology and its application to the Air Traffic Control system are described. The advantages of applying ASR to Air Traffic Control, as well as criteria for choosing a suitable ASR system are presented. Results from previous research and directions for future work at the Flight Transportation Laboratory are outlined.

Introduction

M.I.T.'s Flight Transportation Laboratory (FTL) is renewing its research on the application of Automatic Speech Recognition (ASR) technology to Air Traffic Control (ATC). This report presents an overview of the available technology and its potential use within the ATC system. ATC is a suitable candidate for the application of speech input/output technology due to the well defined syntax and existing reliance on voice communication. Other motivations for introducing ASR into the Air Traffic Control environment are listed within the body of this report. Furthermore, past research efforts are described, with emphasis on work already completed by the Flight Transportation Laboratory. Finally, directions for future research are outlined.

Introduction.

- Just what is Automatic Speech Recognition (ASR) anyway?
- ASR in Air Traffic Control.
- Some motivations for using ASR in Air Trafic Control.
- Previous work.
- Conclusions from Trikas' work.
- Work to be done at the Flight Transportation Laboratory.

Automatic Speech Recognition

ASR systems consist of hardware and software that convert verbal input into machine-useable form (i.e. "text"). These systems can be categorized by three basic parameters: Speaker dependence/independence describes whether the system has to be trained by the user before operational use (speaker dependent), or whether it can be used by any user without specific training (speaker independent). Discrete/connected/continuous speech recognition describes the extent to which naturally spoken speech can be recognized. Single-utterance (isolated-speech) recognizers impose severe constraints on the user's manner of speech, but are relatively easy to implement. Connected speech recognizers allow the user to speak at a normal rate, but finite pauses must be inserted between each word. A continuous speech system recognizes input spoken at a natural rate, with no artificial pauses. Finally, the number of words that the system can recognize at any time (active vocabulary size) is a critical application and performance parameter.

Automatic Speech Recognition.

An Automatic Speech Recognition (ASR) system, is a system that recognizes verbal input and translates it into text. There are three basic factors that categorize an ASR system:

- Speaker dependence/independence.
- Discrete, connected, or continuous speech recognition.
- Vocabulary size.

ASR in Air Traffic Control

Today, the Air Traffic Control system relies on verbal communication between the air traffic controllers and the pilots of the aircraft in the controlled airspace. Although a computer system exists that processes radar and other information regarding the aircraft, the information contained within the verbal communications is not retained. The introduction of ASR technology would allow this information to be captured. The purpose of this research effort is to demonstrate the feasibility of using ASR technology within the ATC environment, and to address the problems involved, especially the relevant human factors issues. Off-the-shelf ASR technology will be used in conjunction with FTL's real-time ATC simulator running on the laboratory's TI-Explorer Lisp machines.

ASR in Air Traffic Control.

We want the "computer" to capture the information given by the controller to aircraft, so that it can be processed. In this particular project, we want to start by using ASR to drive the Flight Transportation Laboratory's real-time ATC simulator.

Why use ASR in ATC?

There are several strong motivations for introducing speech input/output technology into the Air Traffic Control system. Communications are already in the verbal form, and the syntax used is clearly defined by the FAA, and has to some degree been designed to reduce the possibility of communication errors. The use of voice as an input modality allows for a high information throughput capacity, and allows the controllers to keep their eyes and hands busy controlling traffic. Once the verbal information has been captured, it can be transferred to the aircraft via Mode S, conformance monitoring can be improved, and routine clearances can be pre-stored during less busy periods.

Why use ASR in ATC?

- **ATC communication is verbal.**
- **ATC syntax is clearly defined.**
- **ATC training can be automated.**
- **High information throughput.**
- **ASR allows controller to use hands and eyes where they belong.**
- **Captured information can be transmitted to aircraft via Mode S.**
- **Conflict alert can be improved.**
- **Clearances can be pre-stored.**

Previous research.

ASR technology can be used in many aviation and non-aviation applications, and as a result, much research has been conducted on the use of speech input/output in general. However, relatively little research has been dedicated towards the application of ASR to Air Traffic Control. The research to be undertaken within the framework of this project will be a continuation of the initial work presented in Thanassis Trikas' S.M. thesis, *Automated Speech Recognition in Air Traffic Control* (FTL report R87-2).

Previous research.

A lot of research has been done on ASR, but not much in conjunction with ATC:

- **FTL: Thanassis Trikas S.M. work.**
- **Arthur Gerstenfeld (Worcester Polytechnic Institute/UFA, Inc.): Emphasis on ATC training.**
- **ITT Defense Communications Division VRS 1280 demonstration.**

Trikas' conclusions.

Trikas' thesis demonstrated the feasibility of using ASR technology in conjunction with an ATC simulator, utilizing a relatively small vocabulary. An initial error correction strategy based on verbal correction commands alone proved to be unacceptable. Also, problems related to speech articulation variations were encountered. In the process of evaluating his experiment, Trikas implicitly set forth a set of criteria for selecting a suitable ASR system.

Trikas' conclusions.

Trikas' S.M. thesis was essentially a proof of concept of using ASR in ATC:

- ASR <u>can</u> be used with the ATC simulator (with an active vocabulary of only 64 words).
- Correction of recognition errors using voice alone is not feasible.
- Problems with sensitivity to variations in articulation.
- Developed criteria for choosing a suitable ASR system.

Selecting the right ASR system.

The first step in renewing FTL's ASR research effort will be to select a suitable hardware system. For this purpose, performance criteria specific to ATC applications of speech input/output technology have been defined.

Selecting the right ASR system.

Our particular application calls for the following ASR requirements:

- **Speaker independence <u>not</u> required.**
- **Continuous speech recognition.**
- **Vocabulary size 200-300 words.**
- **95% baseline recognition accuracy.**
- **Well designed training procedure.**
- **Open architecture.**
- **Reduced sensitivity to variations.**
- **Short recognition delays (1-4 s).**

Future work.

The future research to be conducted at FTL will be based on previous work completed by Trikas. Hence, his system set-up must be reactivated. In order to improve the simulation and the overall performance of the system, new hardware will be acquired. The actual research will concentrate on the introduction of multi-modal input, improved error correction and recognition accuracy, the evaluation of Mode S usage, and the application of ASR to secondary functions.

Future work.

- Reassemble Trikas' system.
- Evaluate current ASR technology.
- Acquire a new ASR system.
- Introduce multi-modal input.
- Increase number of commands and responses to improve simulation.
- Improve error checking/correction, as well as recognition accuracy.
- Evaluate Mode S usage.
- Use ASR for functions other than ATC commands.

The Airline Schedule Transition Problem

by

Tsuneo Fujiwara

May 26, 1989

Mr. Fuji's claim

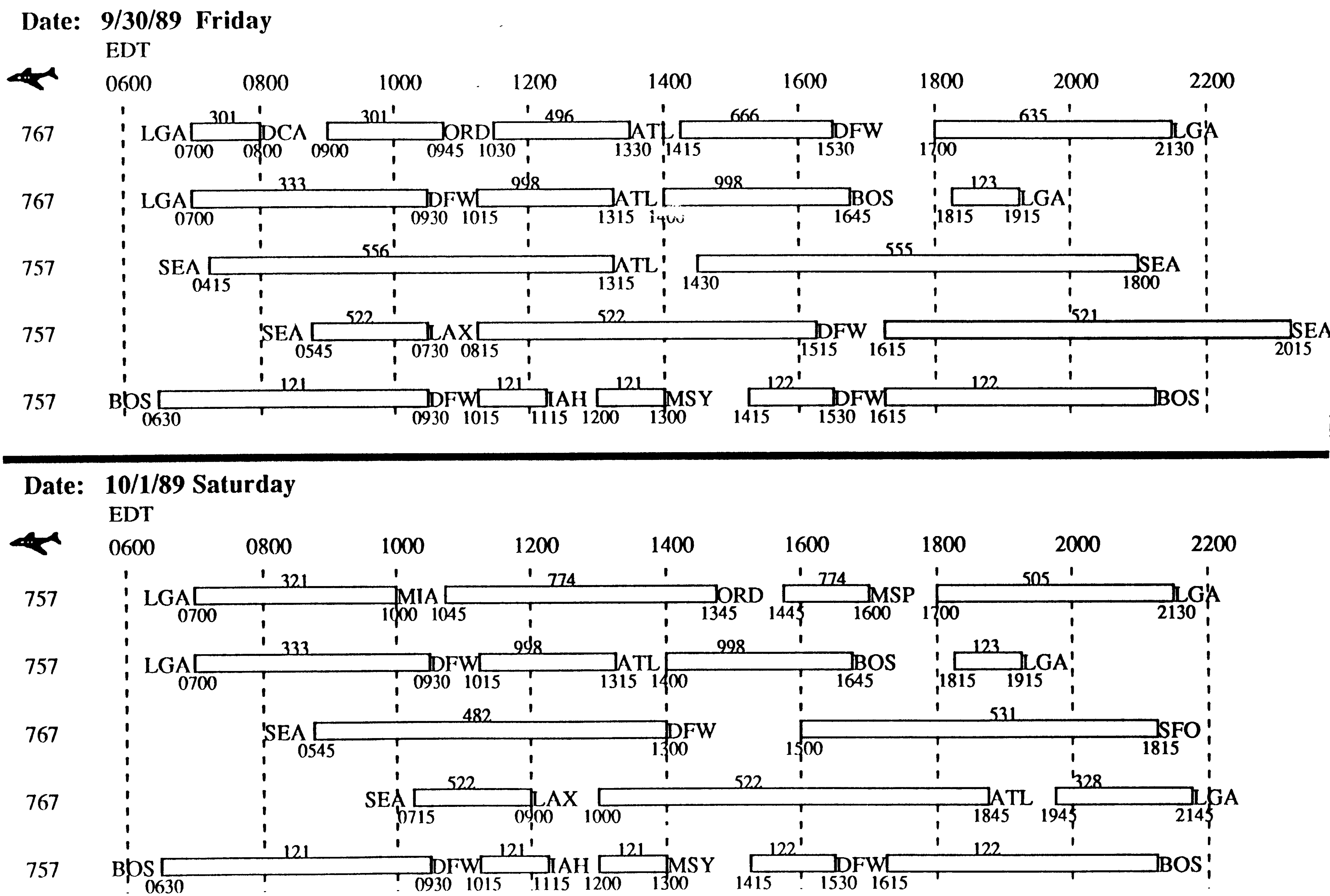
This presentation is less than 30 minutes long!

What is the Airline Schedule Transition Problem ?

Problem of finding the most efficient re-routings of aircraft in order to *balance the number and the types of aircraft at each station* at the beginning of a new schedule.

This presentation is an overview of an attempt to solve this problem automatically for two aircraft types.

Sample Rotation Chart



Important Terms

Aircraft Rotation:

Deficit Station:
The station which has more originating flights of certain aircraft type in the new schedule than terminating flights of that aircraft type in the old schedule.

Flight Schedule:

Flight Segment:

Independent Rotation Pairs:

Intersection:
Two rotations by aircraft of different types are said to intersect at a station if the respective aircraft can be pre-switched/post-switched.

Physical Balance:
The total number of originating flights at a station by all types equals the total number of terminating flights at that station, but their types may be different (e.g. two aircraft type A terminations and one each of aircraft types A and B originating the next morning). However, across all stations served by the fleet, the total number of each terminating aircraft type must match the total number of the corresponding originating aircraft type.

Schedule Transition Period:
The period consisting of n days during which pre-switches and/or post-switches are performed; generally n=2, consisting of the last day of the old schedule and the first day of the new schedule.

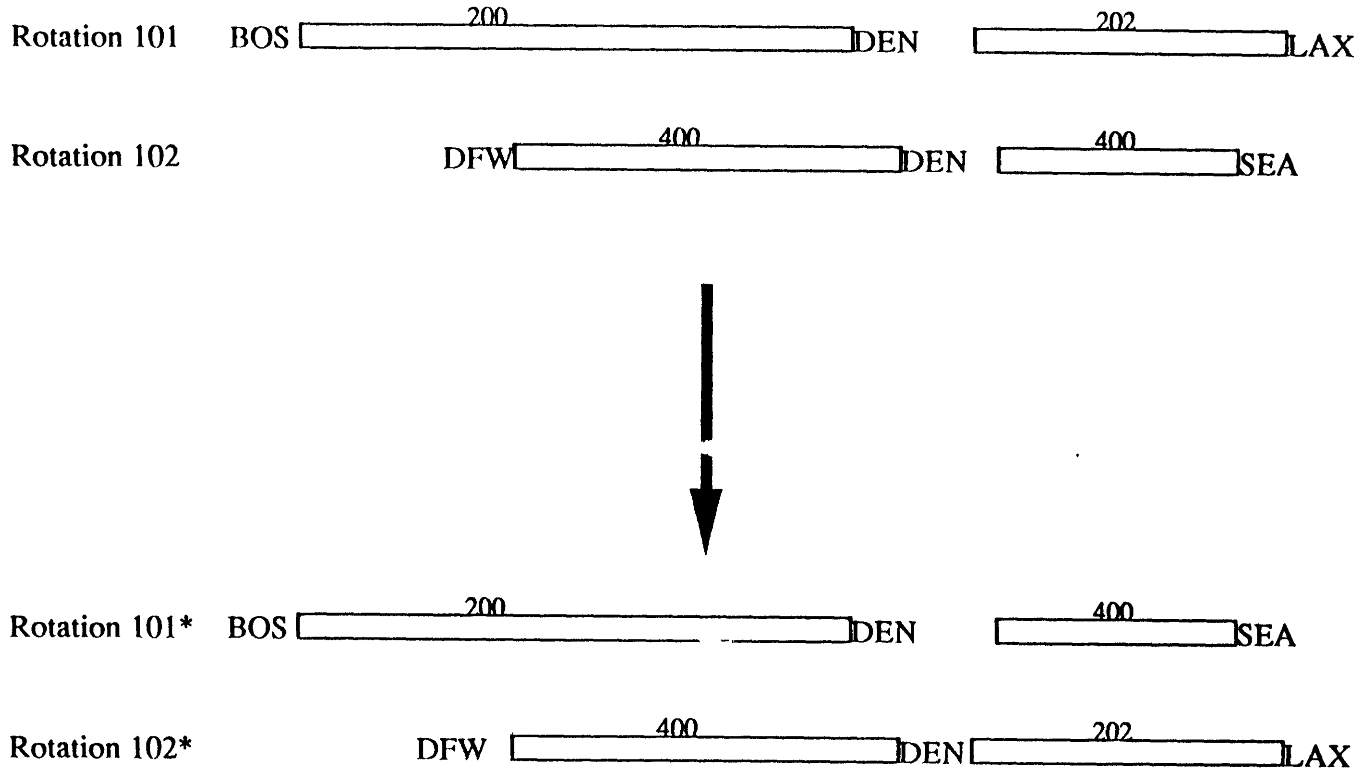
Surplus/Deficit Imbalance Pair:

Surplus Station:
The station which has more terminating flights of certain aircraft type in the old schedule than originating flights of that aircraft type in the new schedule.

Total Balance:
If there are no imbalances at all stations for the period being examined, then the schedule is in total balance.

Turn:
The connection of a specific aircraft tail number from one flight to another.

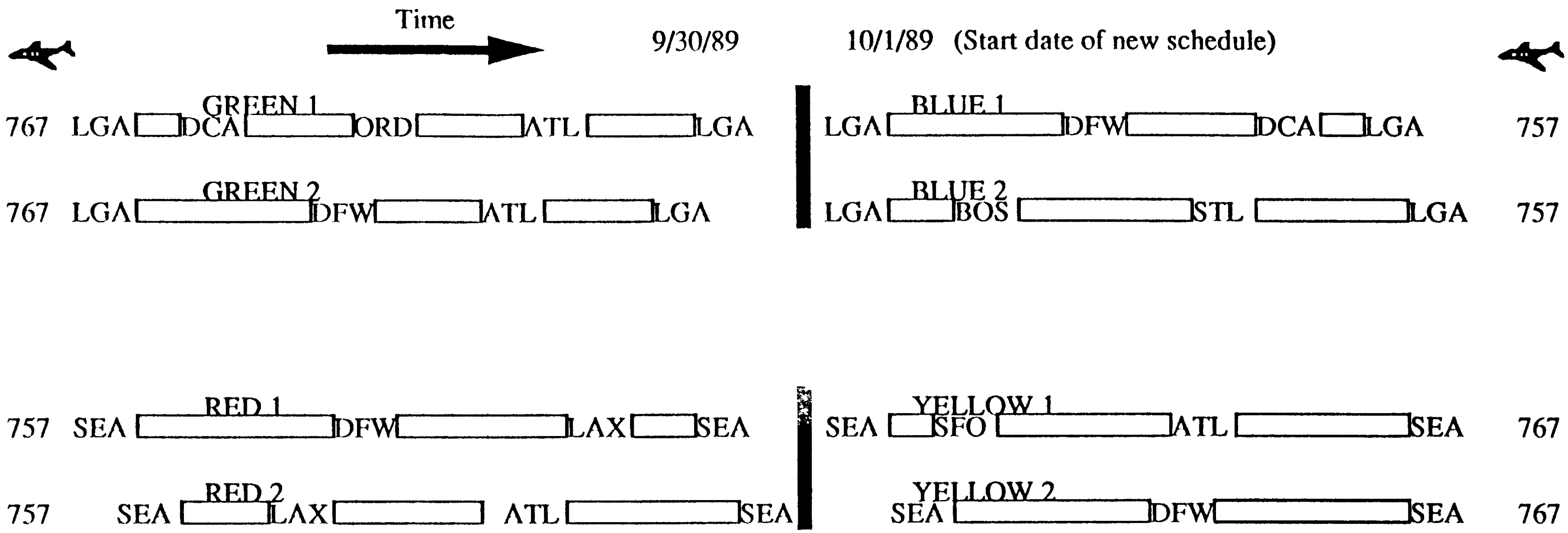
Pre-switch/Post-switch

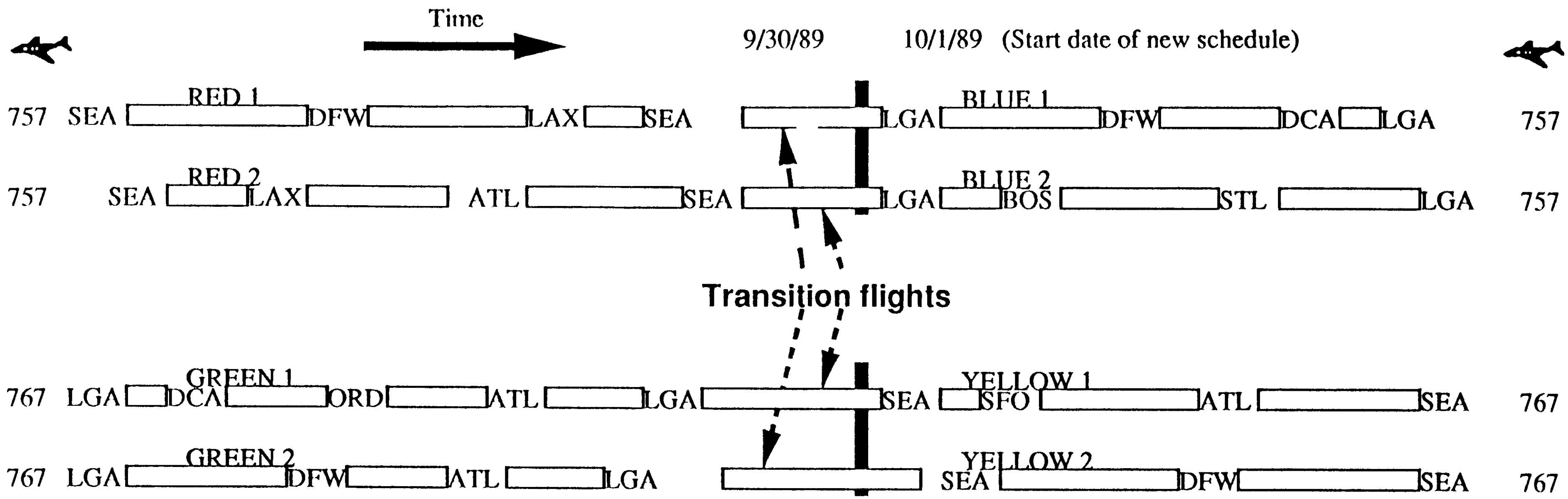


What is the Pre-switch/Post-switch Algorithm ?

It involves *pre-switches* - exchanging aircraft types of certain flights on the last day of the current schedule at some station - and *post-switches* - exchanging aircraft types of certain flights on the first day of the new schedule at some station - to solve the airline schedule transition problem for two aircraft types. It discriminates against *transition flights* - flights that operate only during the schedule transition period to balance the number and/or the types of aircraft needed for the new schedule.

This algorithm can be extended to solve the airline schedule transition problem for more than two aircraft types.

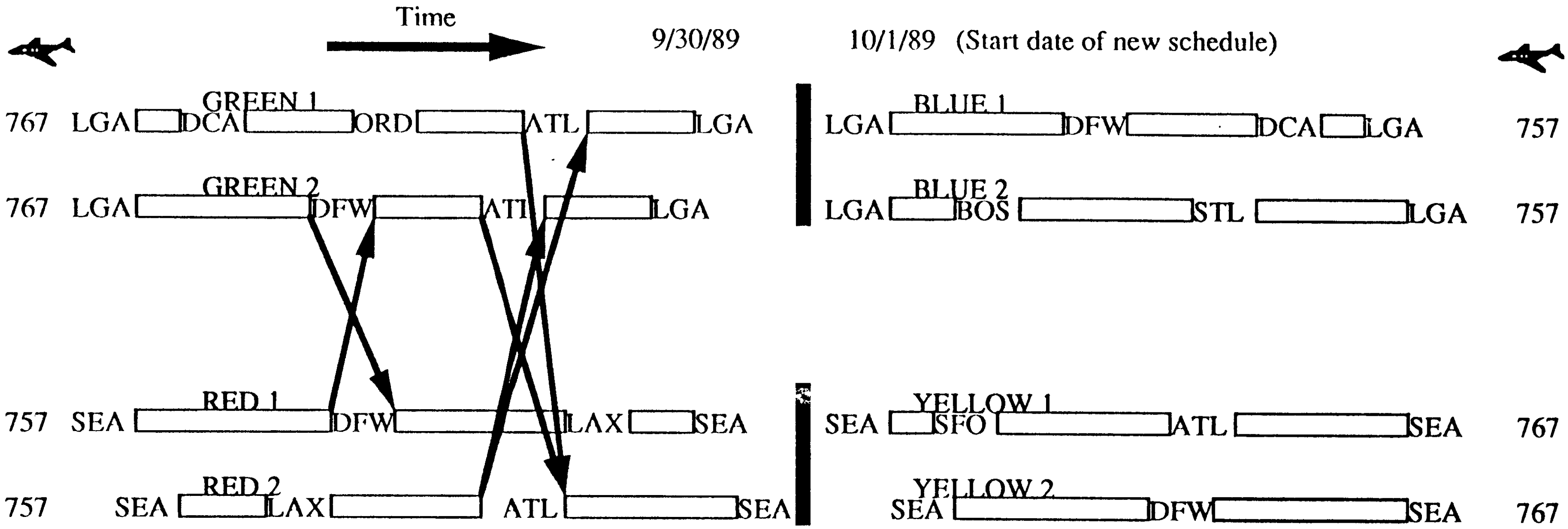

Time
9/30/89
10/1/89 (Start date of new schedule)
GREEN 1
767 LGA DCA ORD ATL LGA
GREEN 2
767 LGA DFW ATL LGA
BLUE 1
LGA DFW DCA LGA 757
BLUE 2
LGA BOS STL LGA 757
RED 1
757 SEA DFW LAX SEA
RED 2
757 SEA LAX ATL SEA
YELLOW 1
SEA SFO ATL SEA 767
YELLOW 2
SEA DFW SEA 767

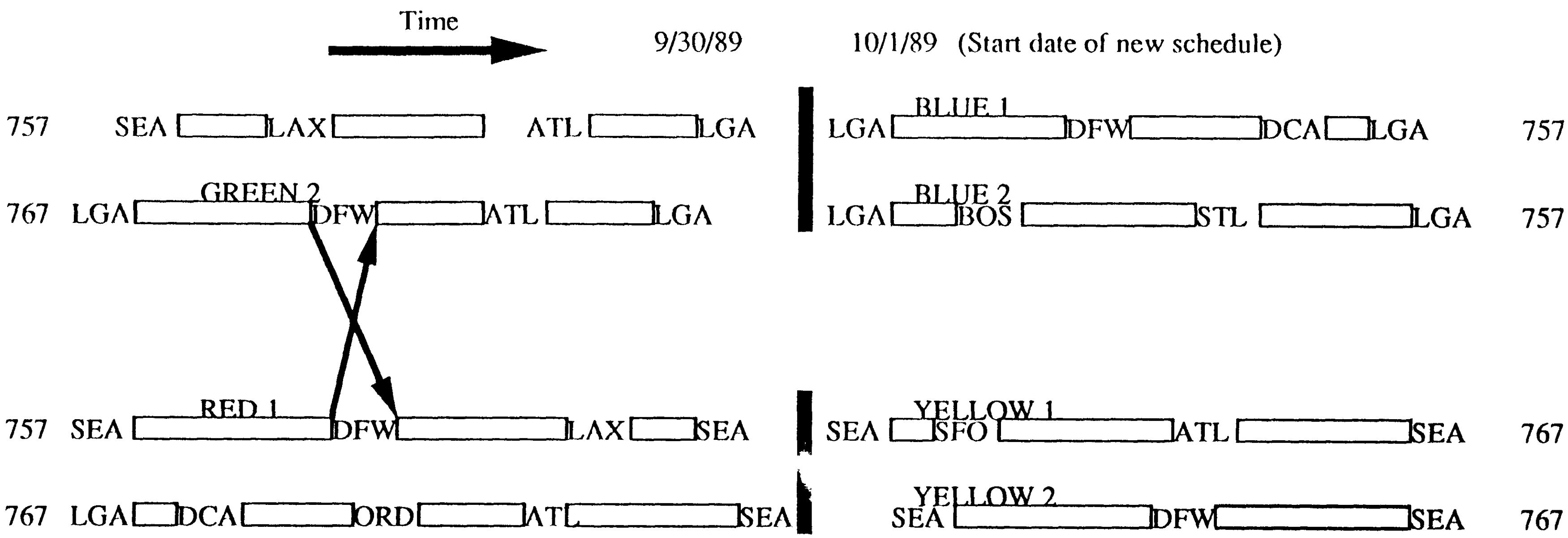


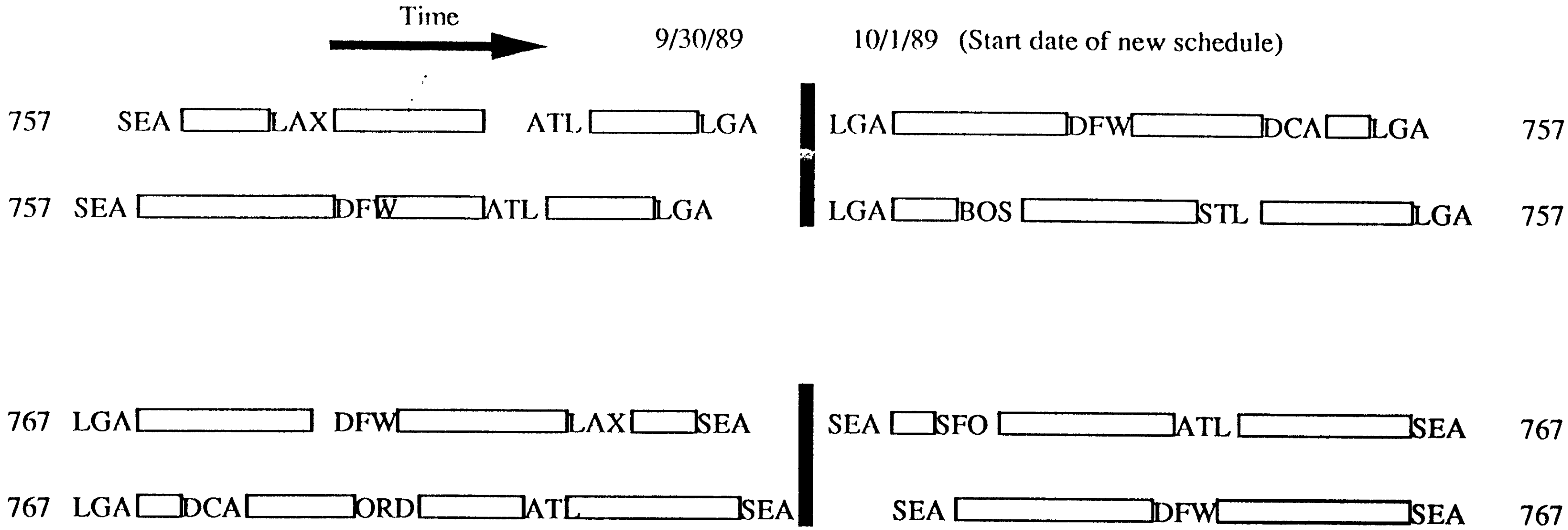
Transition flights are only flown on 9/30/89 to balance the types of aircraft at LGA and SEA.

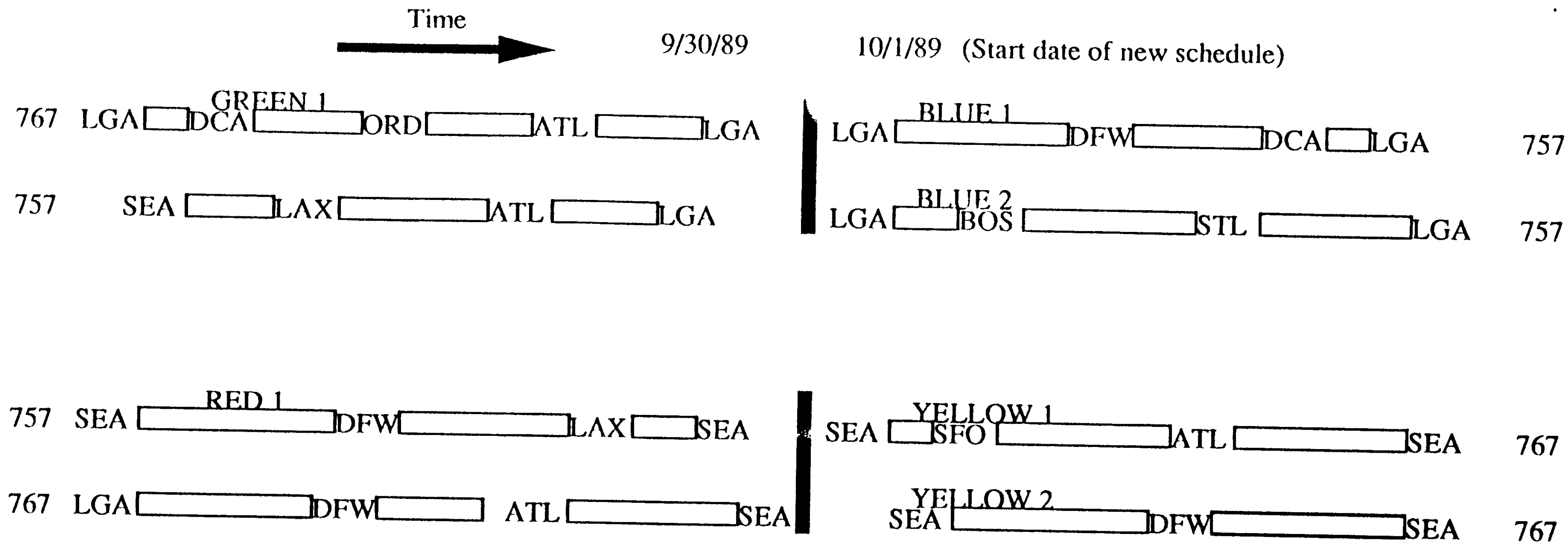
Transition flights are expensive to operate and do not generate much revenue.

Is there a better way to solve the Airline Schedule Transition Problem ?


Time
9/30/89
10/1/89 (Start date of new schedule)
GREEN 1
767 LGA DCA ORD ATL LGA
GREEN 2
767 LGA DFW ATL LGA
RED 1
757 SEA DFW LAX SEA
RED 2
757 SEA LAX ATL SEA
BLUE 1
LGA DFW DCA LGA 757
BLUE 2
LGA BOS STL LGA 757
YELLOW 1
SEA SFO ATL SEA 767
YELLOW 2
SEA DFW SEA 767


Time
9/30/89
10/1/89 (Start date of new schedule)
757 SEA LAX ATL LGA
LGA BLUE 1 DFW DCA LGA 757
767 LGA GREEN 2 DFW ATL LGA
LGA BLUE 2 BOS STL LGA 757
757 SEA RED 1 DFW LAX SEA
SEA YELLOW 1 SFO ATL SEA 767
767 LGA DCA ORD ATL SEA
YELLOW 2 SEA DFW SEA 767

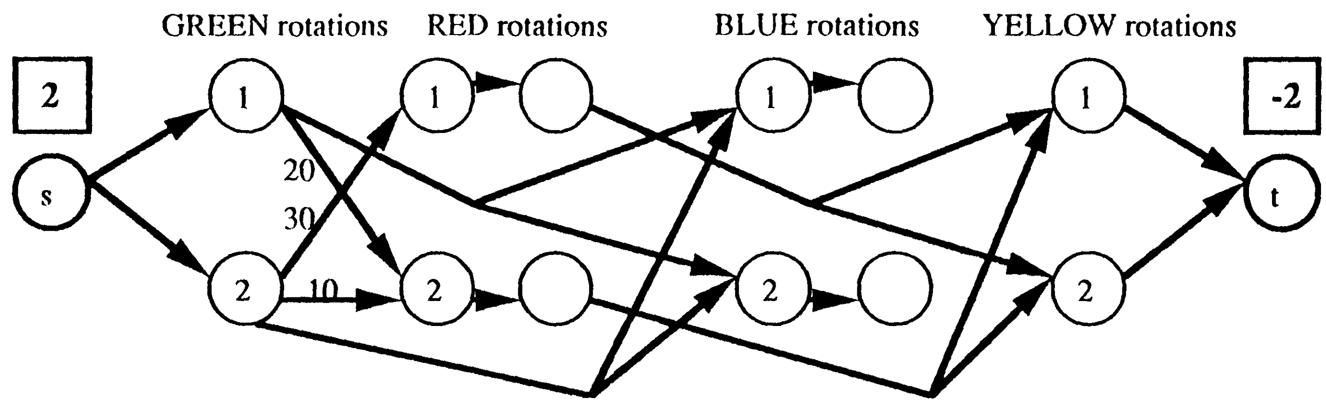

Time
9/30/89
10/1/89 (Start date of new schedule)
757 SEA LAX ATL LGA
LGA DFW DCA LGA 757
757 SEA DFW ATL LGA
LGA BOS STL LGA 757
767 LGA DFW LAX SEA
SEA SFO ATL SEA 767
767 LGA DCA ORD ATL SEA
SEA DFW SEA 767


Time
9/30/89
10/1/89 (Start date of new schedule)
767 LGA DCA ORD ATL LGA
GREEN 1
757 SEA LAX ATL LGA
LGA DFW DCA LGA 757
BLUE 1
LGA BOS STL LGA 757
BLUE 2
757 SEA DFW LAX SEA
RED 1
767 LGA DFW ATL SEA
SEA SFO ATL SEA 767
YELLOW 1
SEA DFW SEA 767
YELLOW 2

Some Points of Interest

- Performing one pre-switch (or post-switch) removes one imbalance from two distinct stations which are the terminating (or originating) stations for the rotations involved.
- There is a cost associated with each pre-switch (and post-switch) which is determined from the operating costs of the aircraft types involved and the load factor data for the affected flights.
- Randomly performing pre-switches (and/or post-switches) will not always get you the best results. Intersections are necessary in order to perform a pre-switch/post-switch.
- In general, the fewer the number of pre-switches/post-switches needed, the better.

Can we automatically find the best set of pre-switches/post-switches ?



Each edge has capacity one.

The number associated with some edges is the cost of pre-switches.

The edges without numbers have a cost of zero.

s is a supply node and t is a demand node.

Solving a min-cost flow problem from s to t solves the schedule transition problem.

Pre-switch/Post-switch Algorithm

Step 0: Physically balance all stations by cancelling/adding flight segments at stations with surpluses/deficits (stage 1 of schedule transition problem) of some aircraft types.

Step 1: Find all stations with a surplus of aircraft type A and a deficit of aircraft type B (surplus A/deficit B stations). Color terminating rotations by type A at these stations GREEN. Color originating rotations by type B at these stations BLUE.

Step 2: Find all stations with a surplus of aircraft type B and a deficit of aircraft type A (surplus B/deficit A stations). Color terminating rotations by type B at these stations RED. Color originating rotations by type A at these stations YELLOW.

Step 3: Look for intersections on the last day of the old schedule between flight segments of GREEN rotation(s) and RED rotation(s) or for intersections on the first day of the new schedule between flight segments of BLUE rotation(s) and YELLOW rotation(s). The identified intersections are candidates for pre-switch or post-switch, respectively.

Step 4: By selectively pre-switching and/or post-switching aircraft at the intersections identified, eliminate aircraft imbalances at the lowest possible overall cost.

Time starting current report : 4/24/1989 at 16:21:49

MISSING FLIGHT SEGMENTS IN ROTATION

FLOW BALANCE FOR : 890930 TO 891001

L10

STATION	890930 TERM	891001 ORIG	TOT-T	TOT-O
ATL	2	2	2	2
BDL	1	1	1	1
BOS	2	2	2	2
DFW	1	1	1	1
EWR	2	2	2	2
FLL	1	3	1	3
JFK	1	1	1	1
LAX	2	2	2	2
LGA	1	1	1	1
MCO	2	1	2	1
ORD	2	2	2	2
PBI	1	1	1	1
SEA	1	1	1	1
SFO	1	0	1	0

767

STATION	890930 TERM	891001 ORIG	TOT-T	TOT-O
FLL	2	0	2	0
MCO	0	1	0	1
SFO	0	1	0	1
TPA	1	1	1	1

757

STATION	890930 TERM	891001 ORIG	TOT-T	TOT-O

PHYSICAL IMBALANCE CHECK

STATION TOT TERM TOT ORIG

No entries below this line.

Time starting current report : 4/24/1989 at 16:29: 9

Pre-switch Post-switch FOR : 880930 TO 881001

a/c 1 : L10 a/c 2 : 767

FLL has surplus a/c 2 and deficit a/c 1.
Rotation 2 will be colored RED. Stations visited :
FLL visited after 2740
LGA visited between 2355 and 2455
ATL visited between 2042 and 2155
MCO visited between 1820 and 1925
ATL visited between 1637 and 1707
SFO visited before 1231
Rotation 7 will be colored RED. Stations visited :
FLL visited after 2826
BOS visited between 2420 and 2520
ATL visited between 2103 and 2205
BDA visited between 1555 and 1755
BOS visited between 1325 and 1355
MCO visited before 1056
Rotation 12 will be colored YELLOW. Stations visited :
FLL visited before 934
ATL visited between 1109 and 1139
MIA visited between 1315 and 1415
ATL visited between 1558 and 1650
PBI visited between 1820 and 1920
ATL visited between 2100 and 2155
MCO visited between 2312 and 2412
EWR visited after 2625
Rotation 13 will be colored YELLOW. Stations visited :
FLL visited before 946
MCO visited between 1031 and 1101
DFW visited between 1330 and 1528
SFO visited between 1855 and 2220
DFW visited between 2518 and 2619
MCO visited after 2830
Rotation 14 will be colored YELLOW. Stations visited :
FLL visited before 1142
ATL visited between 1320 and 1350
MCO visited between 1503 and 1605
ATL visited between 1722 and 1752
BDA visited between 2012 and 2050
BOS visited between 2257 and 2344
BDL visited after 2425

MCO has surplus a/c 1 and deficit a/c 2.
Rotation 4 will be colored GREEN. Stations visited :
MCO visited after 2950
ATL visited between 2718 and 2835
EWR visited between 2405 and 2510
ATL visited between 2141 and 2213
MCO visited between 1954 and 2024

BOS visited between 1635 and 1705
PBI visited before 1400
Rotation 7 will be colored BLUE. Stations visited :
MCO visited before 1056
BOS visited between 1325 and 1355
BDA visited between 1555 and 1755
ATL visited between 2103 and 2205
BOS visited between 2420 and 2520
FLL visited after 2826
Rotation 13 will be colored GREEN. Stations visited :
MCO visited after 2830
DFW visited between 2518 and 2619
SFO visited between 1855 and 2220
DFW visited between 1330 and 1528
MCO visited between 1031 and 1101
FLL visited before 946

SFO has surplus a/c 1 and deficit a/c 2.
Rotation 2 will be colored BLUE. Stations visited :
SFO visited before 1231
ATL visited between 1637 and 1707
MCO visited between 1820 and 1925
ATL visited between 2042 and 2155
LGA visited between 2355 and 2455
FLL visited after 2740
Rotation 23 will be colored GREEN. Stations visited :
SFO visited after 2833
DFW visited between 2357 and 2459
FLL visited between 2005 and 2110
ATL visited before 1826

Pre-switch rotation 4 between 2141 and 2213
with rotation 2 between 2042 and 2155 at station ATL.
Pre-switch rotation 4 between 2141 and 2213
with rotation 7 between 2103 and 2205 at station ATL.
Pre-switch rotation 13 between 1031 and 1101
with rotation 7 between 0 and 1056 at station MCO.
Pre-switch rotation 23 between 0 and 1826
with rotation 2 between 1637 and 1707 at station ATL.
Post-switch rotation 7 between 2103 and 2205
with rotation 12 between 2100 and 2155 at station ATL.
Post-switch rotation 7 between 0 and 1056
with rotation 13 between 1031 and 1101 at station MCO.
Post-switch rotation 2 between 1637 and 1707
with rotation 12 between 1558 and 1650 at station ATL.
Post-switch rotation 2 between 2042 and 2155
with rotation 12 between 2100 and 2155 at station ATL.

The result from Automatic Pre-switch/Post-switch algorithm :
Pre-switch rotations 13 and 7 at 1031 and 0
Pre-switch rotations 23 and 2 at 0 and 1637

No entries below this line.

Other Related Topics

- Multiple aircraft type schedule transition problem
  - -Simple Approach
  - -Greedy Approach
  - -All-at-once Approach
  - -Set-covering Approach

- Integrating intermediate pre-switches/post-switches

- Totally balancing holiday period and daily flight schedules using Pre-switch/Post-switch Algorithm

The End

COMPARISON OF YIELD MANAGEMENT STRATEGIES: SIMULATION RESULTS

Dr. Peter P. Belobaba

Flight Transportation Laboratory
Massachusetts Institute of Techr·ology
Cambridge, MA 02139

Presentation to MIT/FTL
Cooperative Research Program Annual Meeting

May 25, 1989

OUTLINE

1. Terminology and Definitions

2. Booking Class Structures

3. Optimization Methods

4. Simulation Results

5. Conclusions

1. TERMINOLOGY AND DEFINITIONS

A. INVENTORY STRUCTURES

BOOKING CLASS (also FARE CLASS): Each fare basis code is associated with a booking class; booking class availability is displayed on a computer reservations system (CRS) screen.

BOOKING LIMIT (AUTHORIZED LIMIT): The maximum number of bookings that may be accepted in a booking class.

B. CONTROL MECHANISMS

BY BOOKING CLASS (FLIGHT LEG): Limits are applied only to booking classes at the flight leg level.

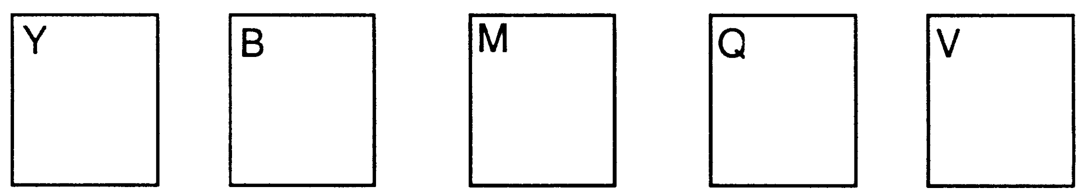
BY ON-FLIGHT ITINERARY(SEGMENT): Within each booking class, additional limits are placed on local vs. through passengers.

BY TOTAL ITINERARY (O-D): Limits are applied to each specific passenger O-D and booking class.

2. BOOKING CLASS STRUCTURES

DISTINCT (PARTITIONED) CLASSES or "BUCKETS":
Each booking class has its own allocation of seats;
allocations sum to capacity of shared cabin.

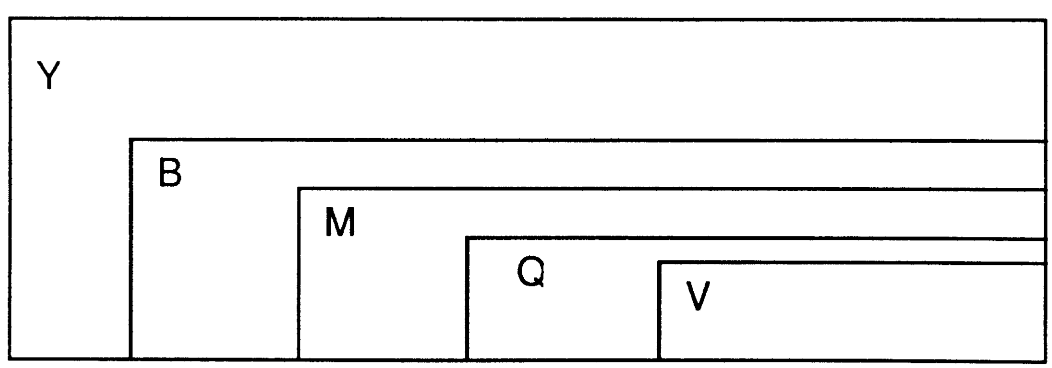
EXAMPLE: 100 seats, 5 classes
Y15 B20 M25 Q30 V10



NESTED CLASSES: Maximum limits are applied to each booking class; each higher class has a higher booking limit.

EXAMPLE: Y100 B85 M65 Q40 V10

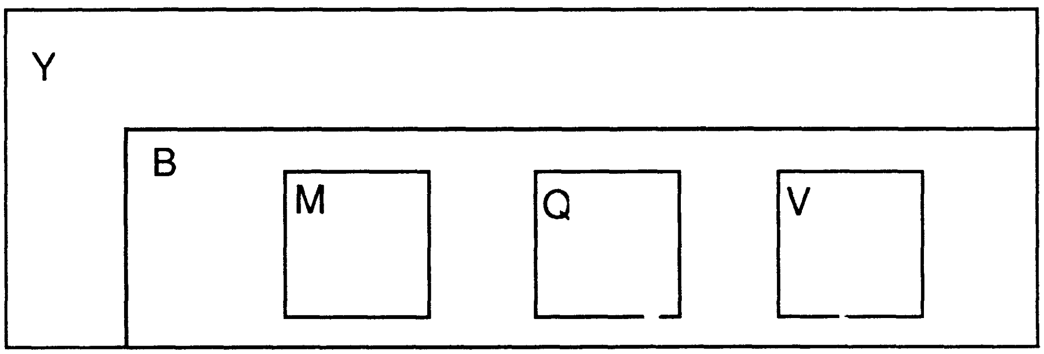
15 seats are <u>protected</u> for exclusive use of Y-class bookings, but Y-class can book up to 100.



PARTIALLY NESTED (HYBRID) CLASSES: Any combination of distinct and nested booking classes.

EXAMPLE: Y100 B85 M25 Q30 V10

Y and B are parent classes; M,Q,V are distinct and nested within B.



3. OPTIMIZATION METHODS

DISTINCT BOOKING CLASSES:

1. DETERMINISTIC ALLOCATION - Allocate seats to each booking class based on mean forecast demands (starting at highest class).

EXAMPLE: 5 Fare Classes, 195 Seats

	CLASS 1	2	3	4	5
Demand	28	43	54	49	46
Std. Error	9.8	15.1	18.9	17.2	16.1
Fare	$289	$236	$205	$141	$127
Deterministic Allocation	28	43	54	49	21

2. PROBABILISTIC OPTIMIZATION - Allocate seats based on probabilistic distribution of forecasted demands, such that expected marginal revenue from last seats allocated is equal across all classes.

	CLASS 1	2	3	4	5
Probabilistic Allocation	31	45	53	37	29

NESTED BOOKING CLASSES:

1. DETERMINISTIC PROTECTION - Protect seats for each booking class based on mean forecast demands, from highest class down.

	CLASS 1	2	3	4	5
Deterministic Protection	28	43	54	49	21

2. ADAPTED PROBABILISTIC - Protect seats for each booking class by applying probabilistic optimal allocations from distinct class problem.

	CLASS 1	2	3	4	5
Adapted Probabilistic	31	45	53	37	29

3. EMSR ALGORITHM (Belobaba 1987) - Find optimal protection limits between each pair of classes and nest the results based on expected marginal seat revenues.

	CLASS 1	2	3	4	5
EMSR Nested Protection	19	31	64	34	47

4. OPTIMAL SOLUTION (Wollmer 1988) - Calculate optimal nested booking limits by considering joint probability distribution of all classes at the same time.

	CLASS 1	2	3	4	5
Optimal Nested Protection	19	36	65	43	32

4. SIMULATION OF FLIGHT LEG CONTROL

INPUTS:

- 5 booking classes on single flight leg
- Probabilistic demand distributions for each class, summing to a mean total demand of 220 (see Table 1)
- Constant, hierarchical class revenues
- Varying capacities, from 100 to 300 (Demand factors from 0.73 to 2.20).
- 3 demand scenarios:
  - (1) Distributed class demands
  - (2) High high-fare demand
  - (3) High low-fare demand

SIMULATION:

- Demands drawn from normal density for each class
- Lowest class books first; highest last
- Independent class demands; no "sell-up"
- Single point in time; no revisions of booking limits during reservations process.
- 10,000 flight sample for each scenario

OUTPUTS:

- Expected flight leg revenues for each demand scenario, capacity and set of booking limits.

TABLE 1: SIMULATION INPUT DATA - FLIGHT LEG CONTROL

SCENARIO 1: DISTRIBUTED FARE CLASS DEMANDS

CLASS	FARE	DEMAND FORECAST	STD ERROR
1	$289	28	9.8
2	$236	43	15.1
3	$205	54	18.9
4	$141	49	17.2
5	$127	46	16.1
	TOTAL	220	

SCENARIO 2: HIGH HIGH-FARE DEMAND

CLASS	FARE	DEMAND FORECAST	STD ERROR
1	$289	35	12.3
2	$236	56	19.6
[illegible]	$205	64	22.4
4	$141	42	14.7
5	$127	23	8.1
	TOTAL	220	

SCENARIO 3: HIGH LOW-FARE DEMAND

CLASS	FARE	DEMAND FORECAST	STD ERROR
1	$289	20	7
2	$236	30	10.5
3	$205	44	15.4
4	$141	56	19.6
5	$127	70	24.5
	TOTAL	220	

AIRCRAFT CAPACITIES

CAPACITY	300	260	230	195	160	130	100
DEMAND FACTOR (Demand/Capacity)	0.73	0.85	0.96	1.13	1.38	1.69	2.20

SIMULATION RESULTS - FLIGHT LEG CONTROL

DISTINCT BOOKING CLASSES

FIGURE 1 shows percentage difference in revenues for deterministic and probabilistic solutions over no control for all 3 demand scenarios:

- At demand factors above 1.13, both solutions show major revenue gains over no control.
- At lower demand factors, use of distinct booking classes actually causes revenue shortfall compared to no control, regardless of solution method.
- In all cases, probabilistic solution outperforms deterministic solution, more so at lower demand factors.

FIGURE 2 shows this comparison of probabilistic over deterministic distinct class solutions for all 3 demand scenarios:

- Probabilistic is 13% higher in expected revenues at demand factor 0.73%.
- Difference decreases rapidly then increases again, especially for unequal fare class distributions of demand (Scenarios 2 and 3).

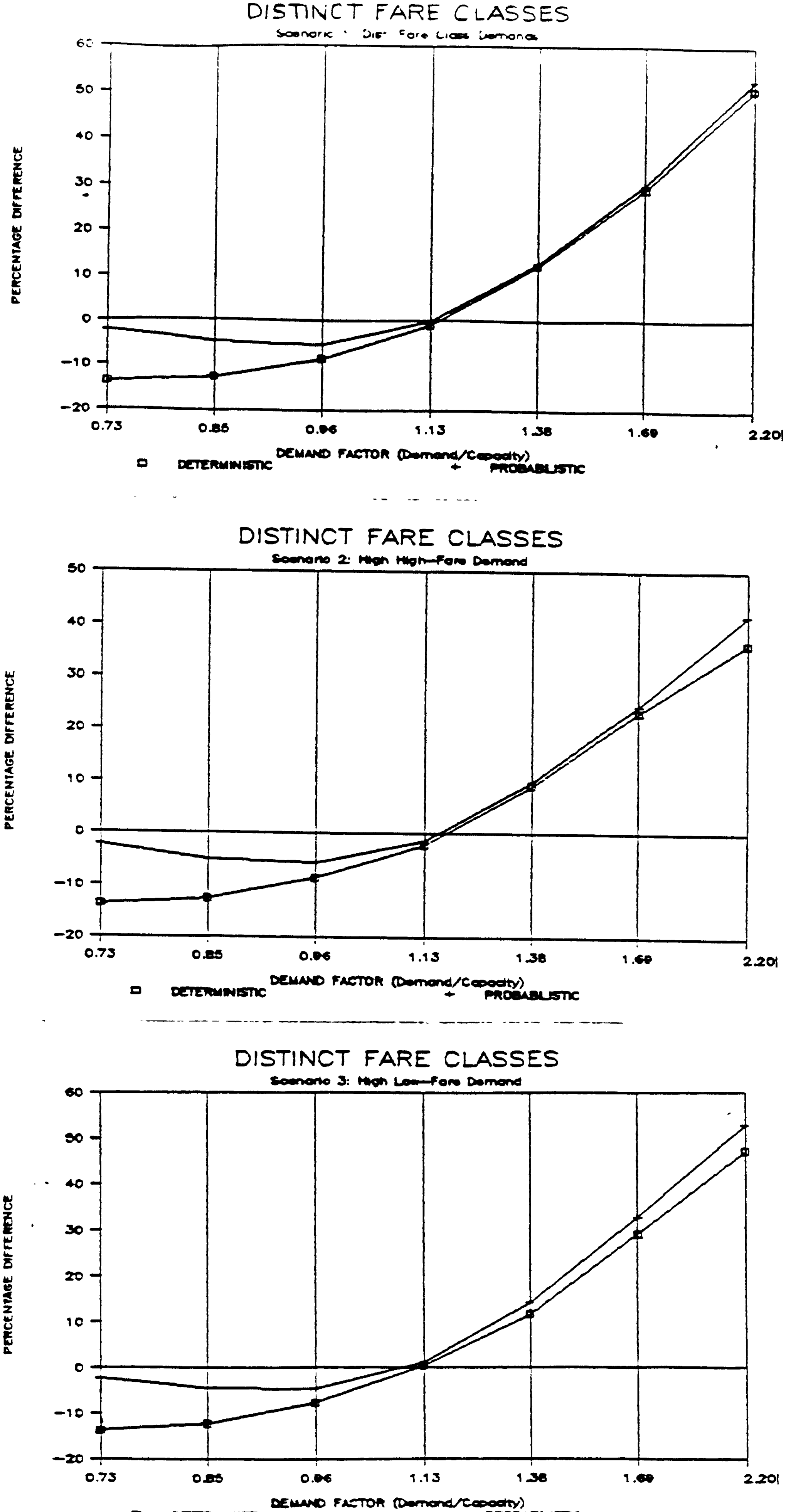

DISTINCT FARE CLASSES
Scenario 1: Dist Fare Class Demand
PERCENTAGE DIFFERENCE
60
50
40
30
20
10
0
-10
-20
0.73
0.85
0.96
1.13
1.38
1.69
2.20
DEMAND FACTOR (Demand/Capacity)
DETERMINISTIC
PROBABLISTIC
DISTINCT FARE CLASSES
Scenario 2: High High-Fare Demand
PERCENTAGE DIFFERENCE
50
40
30
20
10
0
-10
-20
0.73
0.85
0.96
1.13
1.38
1.69
2.20
DEMAND FACTOR (Demand/Capacity)
DETERMINISTIC
PROBABLISTIC
DISTINCT FARE CLASSES
Scenario 3: High Low-Fare Demand
PERCENTAGE DIFFERENCE
60
50
40
30
20
10
0
-10
-20
0.73
0.85
0.96
1.13
1.38
1.69
2.20
DEMAND FACTOR (Demand/Capacity)
DETERMINISTIC
PROBABLISTIC

FIGURE 1

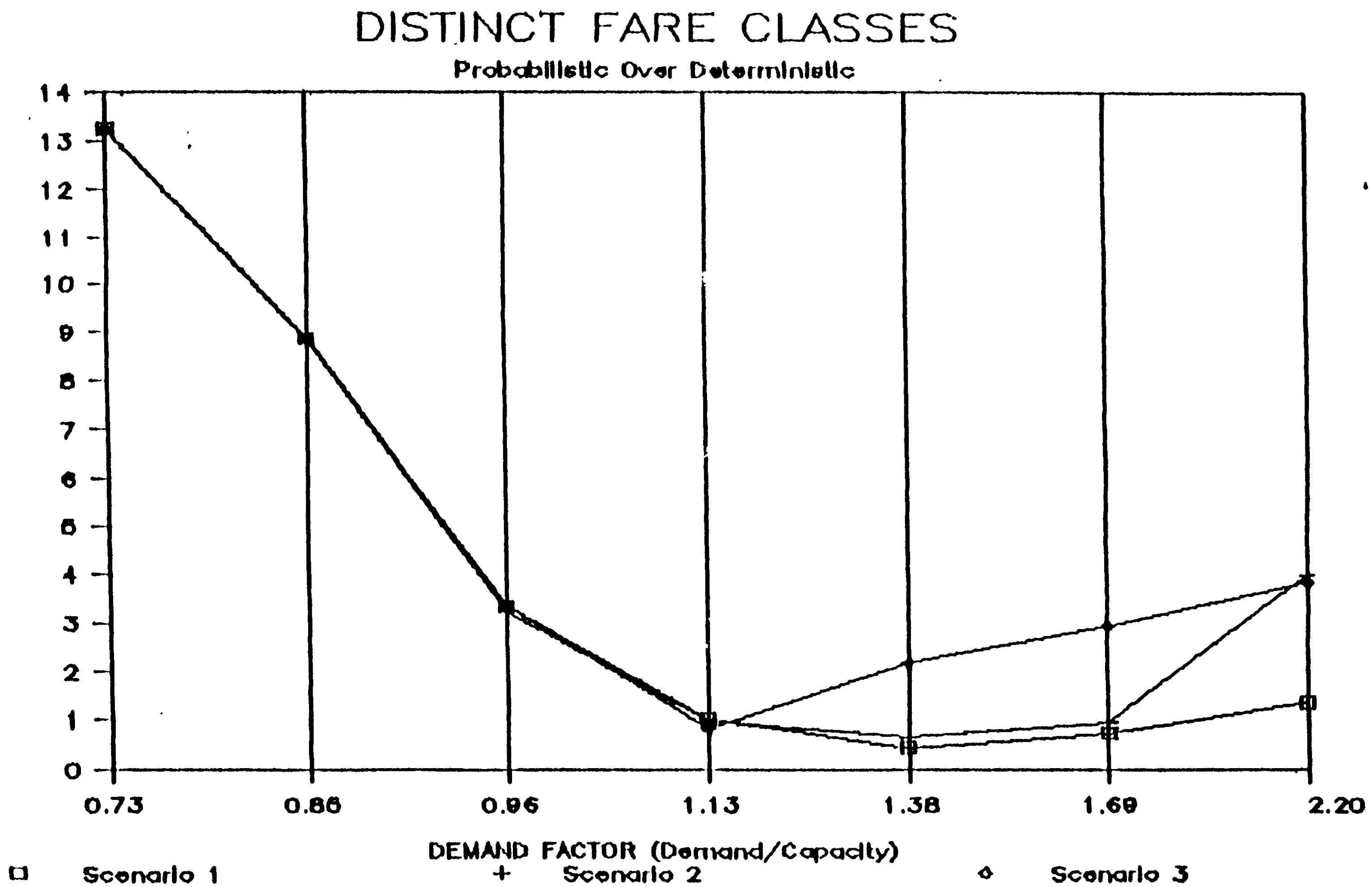

DISTINCT FARE CLASSES
Probabilistic Over Deterministic
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
0.73
0.86
0.96
1.13
1.38
1.69
2.20
DEMAND FACTOR (Demand/Capacity)
Scenario 1
Scenario 2
Scenario 3

FIGURE 2

-156-

SIMULATION RESULTS - FLIGHT LEG CONTROL

NESTED BOOKING CLASSES

FIGURE 3 shows revenue difference over no control for deterministic, adapted probabilistic, and EMSR solutions under all 3 demand scenarios:

- Positive revenue impact of all methods is evident at demand factors above 0.96.

- Adapted probabilistic method has negative revenue impacts at demand factors below 1.0, although it outperformed deterministic solution at high demand factors in 2 scenarios

- EMSR solution showed highest revenues in all cases.

FIGURE 4 shows comparison of EMSR over deterministic solution for the 3 demand scenarios:

- EMSR revenues are 0.5 to 1 percent higher at demand factors around 1.0.

- EMSR advantage increases to 2-3 percent at higher demand factors, except Scenario 3 (high low-fare demand).

FIGURE 5 shows revenue difference of optimal nested solution over EMSR algorithm:

- Optimal solution results in expected revenues marginally greater than EMSR method, but difference is less than 0.5 percent all cases.

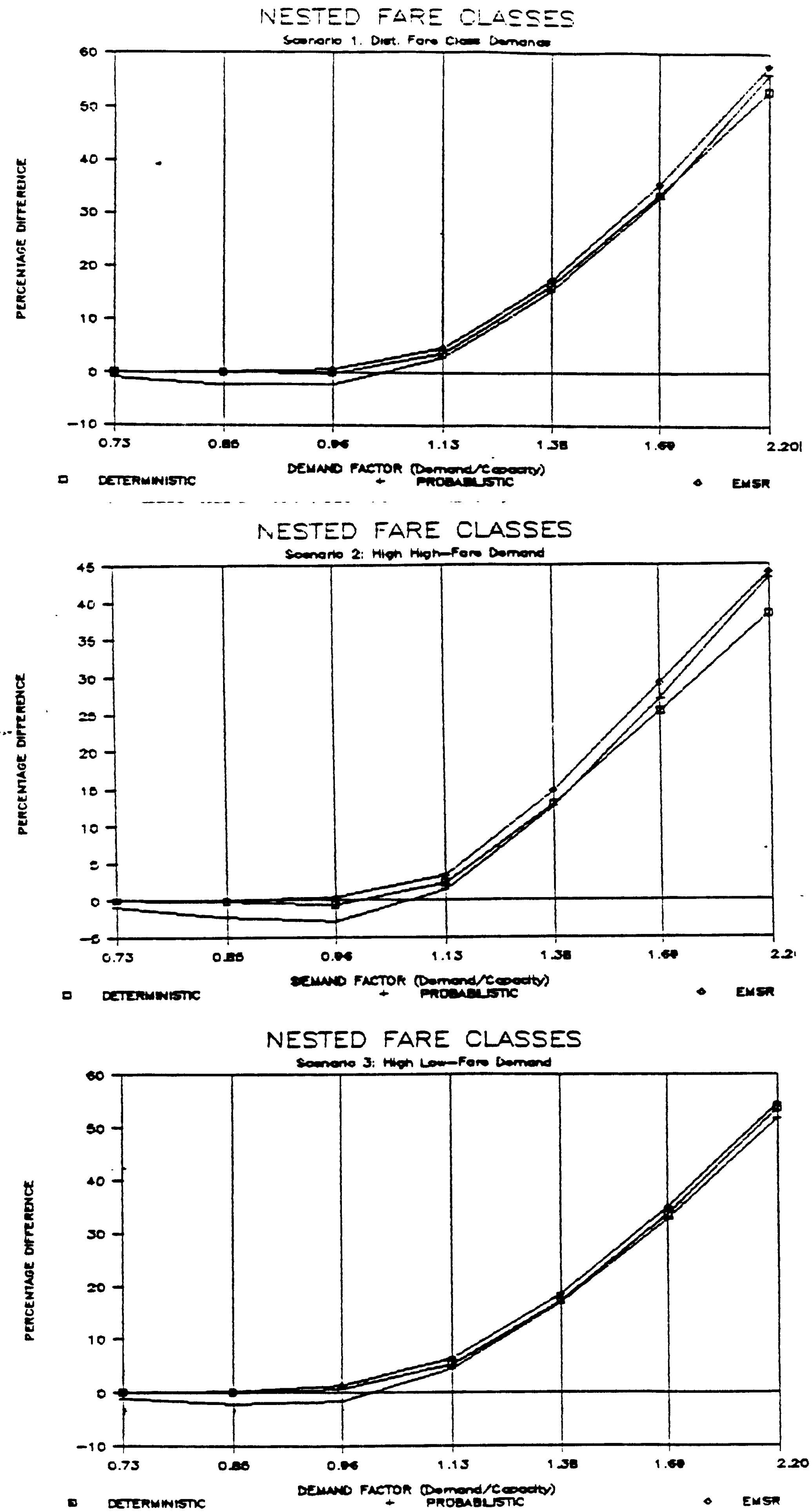
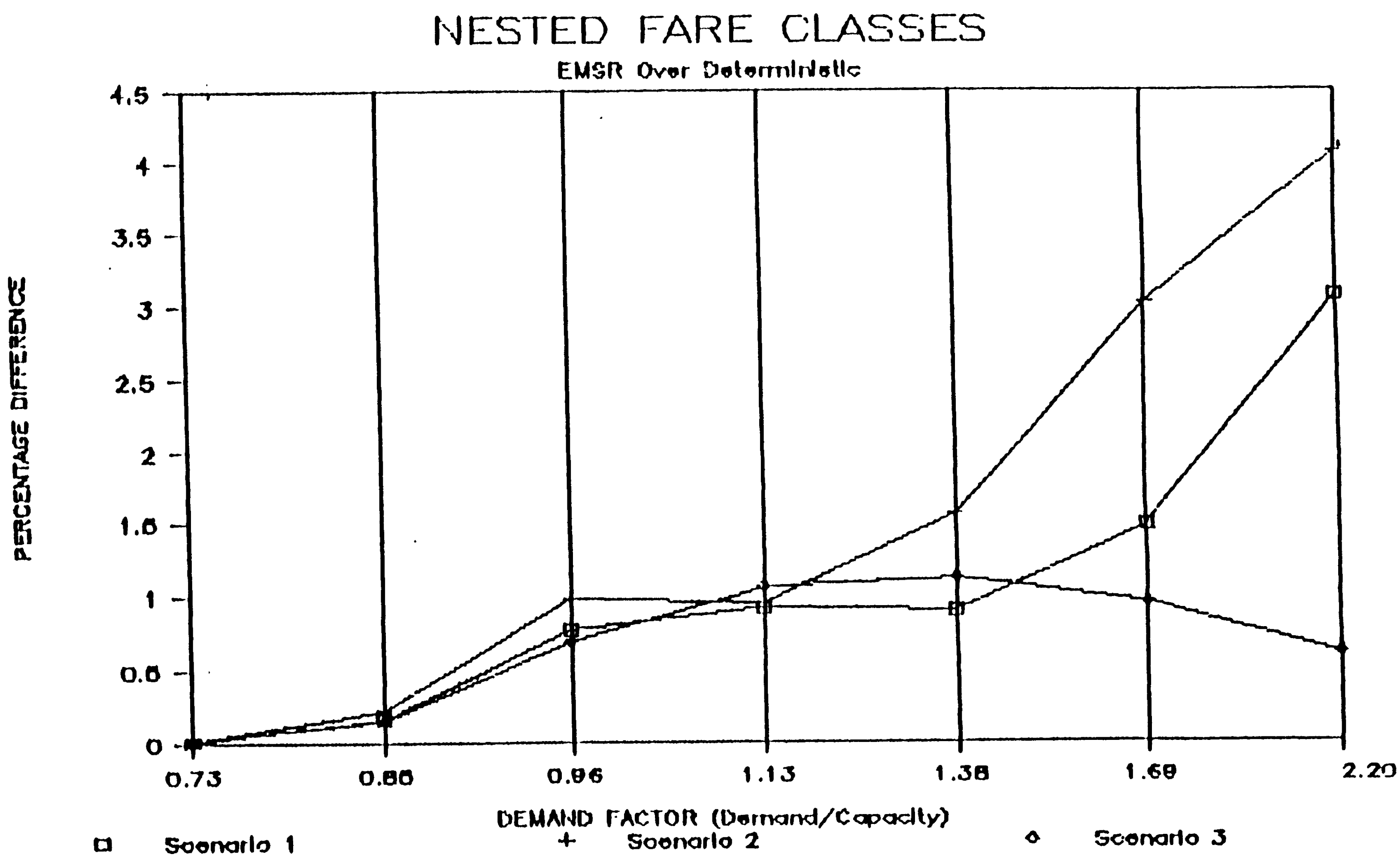

NESTED FARE CLASSES
Scenario 1. Dist. Fare Class Demands
PERCENTAGE DIFFERENCE
60
50
40
30
20
10
0
−10
0.73
0.86
0.96
1.13
1.38
1.69
2.20
DEMAND FACTOR (Demand/Capacity)
DETERMINISTIC
PROBABLISTIC
EMSR
NESTED FARE CLASSES
Scenario 2: High High–Fare Demand
PERCENTAGE DIFFERENCE
45
40
35
30
25
20
15
10
5
0
−5
0.73
0.86
0.96
1.13
1.38
1.69
DEMAND FACTOR (Demand/Capacity)
DETERMINISTIC
PROBABLISTIC
EMSR
NESTED FARE CLASSES
Scenario 3: High Low–Fare Demand
PERCENTAGE DIFFERENCE
60
50
40
30
20
10
0
−10
0.73
0.86
0.96
1.13
1.38
1.69
2.20
DEMAND FACTOR (Demand/Capacity)
DETERMINISTIC
PROBABLISTIC
EMSR

FIGURE 3



NESTED FARE CLASSES
EMSR Over Deterministic
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
PERCENTAGE DIFFERENCE
0.73
0.86
0.96
1.13
1.38
1.69
2.20
DEMAND FACTOR (Demand/Capacity)
Scenario 1
Scenario 2
Scenario 3

FIGURE 4

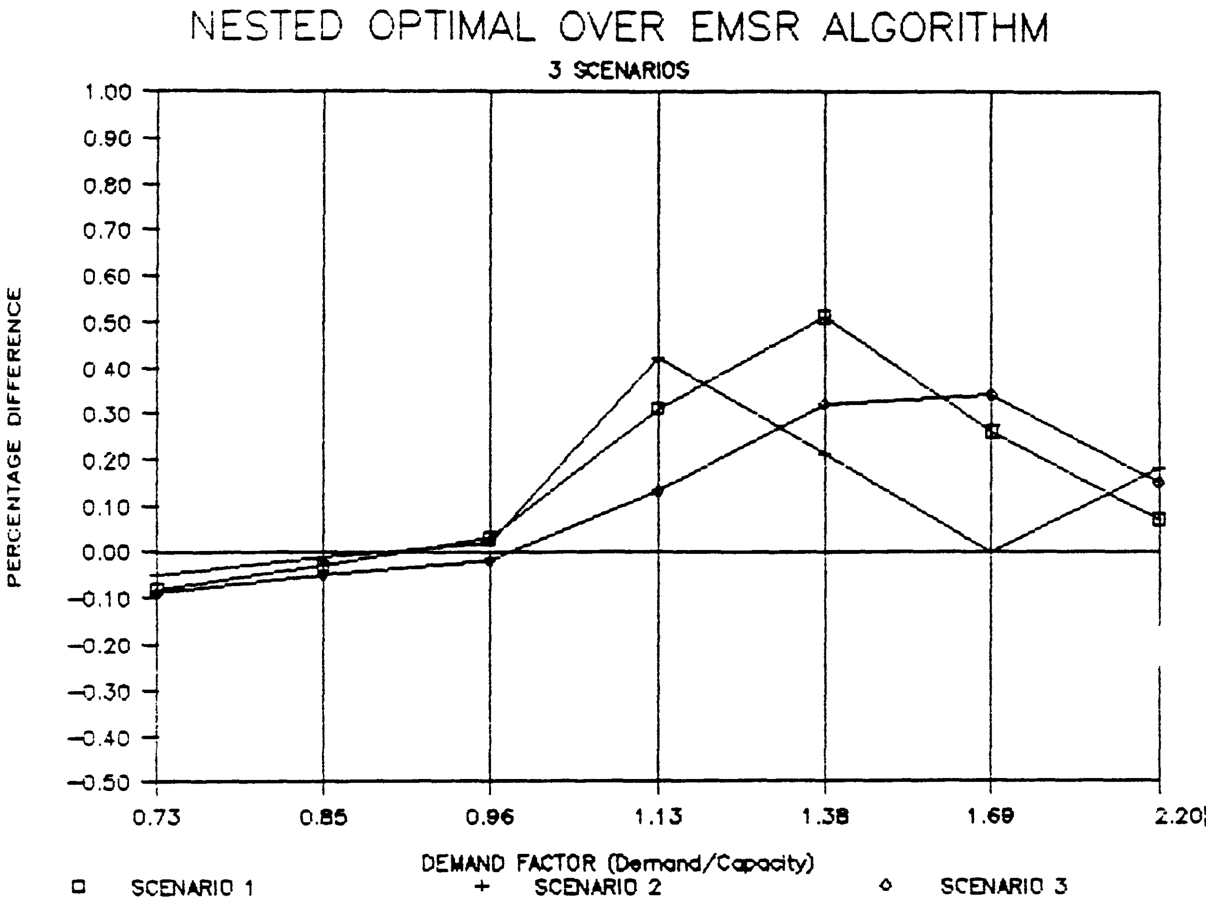

NESTED OPTIMAL OVER EMSR ALGORITHM
3 SCENARIOS
PERCENTAGE DIFFERENCE
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
-0.10
-0.20
-0.30
-0.40
-0.50
0.73
0.85
0.96
1.13
1.38
1.69
2.20
DEMAND FACTOR (Demand/Capacity)
SCENARIO 1
SCENARIO 2
SCENARIO 3

FIGURE 5

FLIGHT LEG SIMULATION RESULTS - SUMMARY

FIGURES 6, 7 and 8 compare expected revenues across all the flight leg options tested for selected demand factors(0.96, 1.13 and 1.38, respectively):

- At demand factor 0.96, distinct methods show substantial negative impact compared to no control. EMSR and optimal nested solution show small positive impacts.

- At demand factor 1.13, all nested methods have positive revenue impacts. Distinct methods show small positive impact for Scenario 3.

- At demand factor 1.38, all methods show positive revenue impact of 9-18 percent over no control. Relative rankings are consistent across scenarios.

FIGURE 9 illustrates positive revenue impact of nested fare classes/EMSR solution over distinct fare classes/probabilistic solution. Nested EMSR revenues are generally at least 2 percent higher, peaking at 6 percent higher for demand factor 0.96.

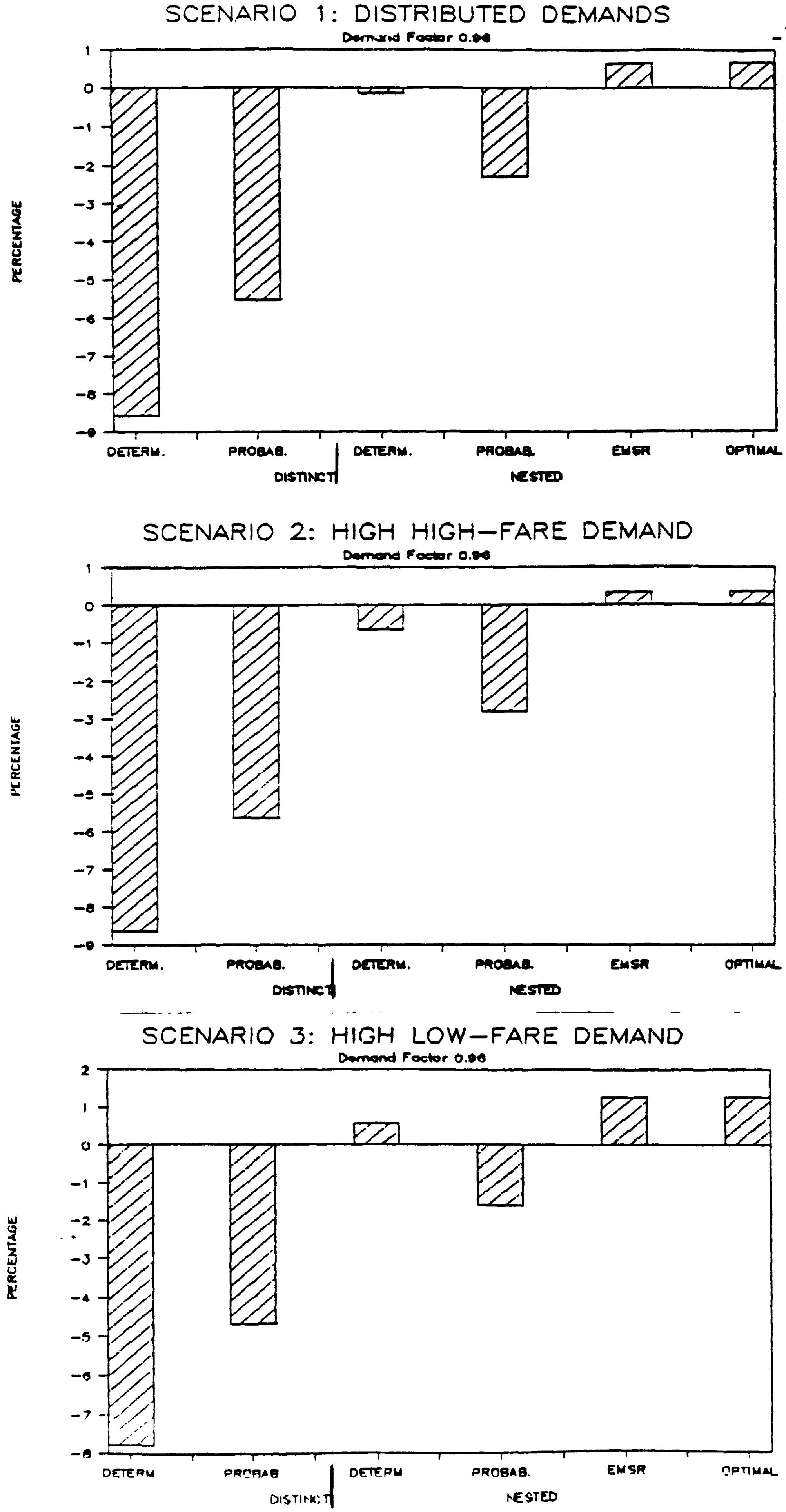

SCENARIO 1: DISTRIBUTED DEMANDS
Demand Factor 0.96
PERCENTAGE
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
DETERM.
PROBAB.
DETERM.
PROBAB.
EMSR
OPTIMAL
DISTINCT
NESTED
SCENARIO 2: HIGH HIGH-FARE DEMAND
Demand Factor 0.96
PERCENTAGE
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
DETERM.
PROBAB.
DETERM.
PROBAB.
EMSR
OPTIMAL
DISTINCT
NESTED
SCENARIO 3: HIGH LOW-FARE DEMAND
Demand Factor 0.96
PERCENTAGE
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
DETERM
PROBAB
DETERM
PROBAB.
EMSR
OPTIMAL
DISTINCT
NESTED

FIGURE 6

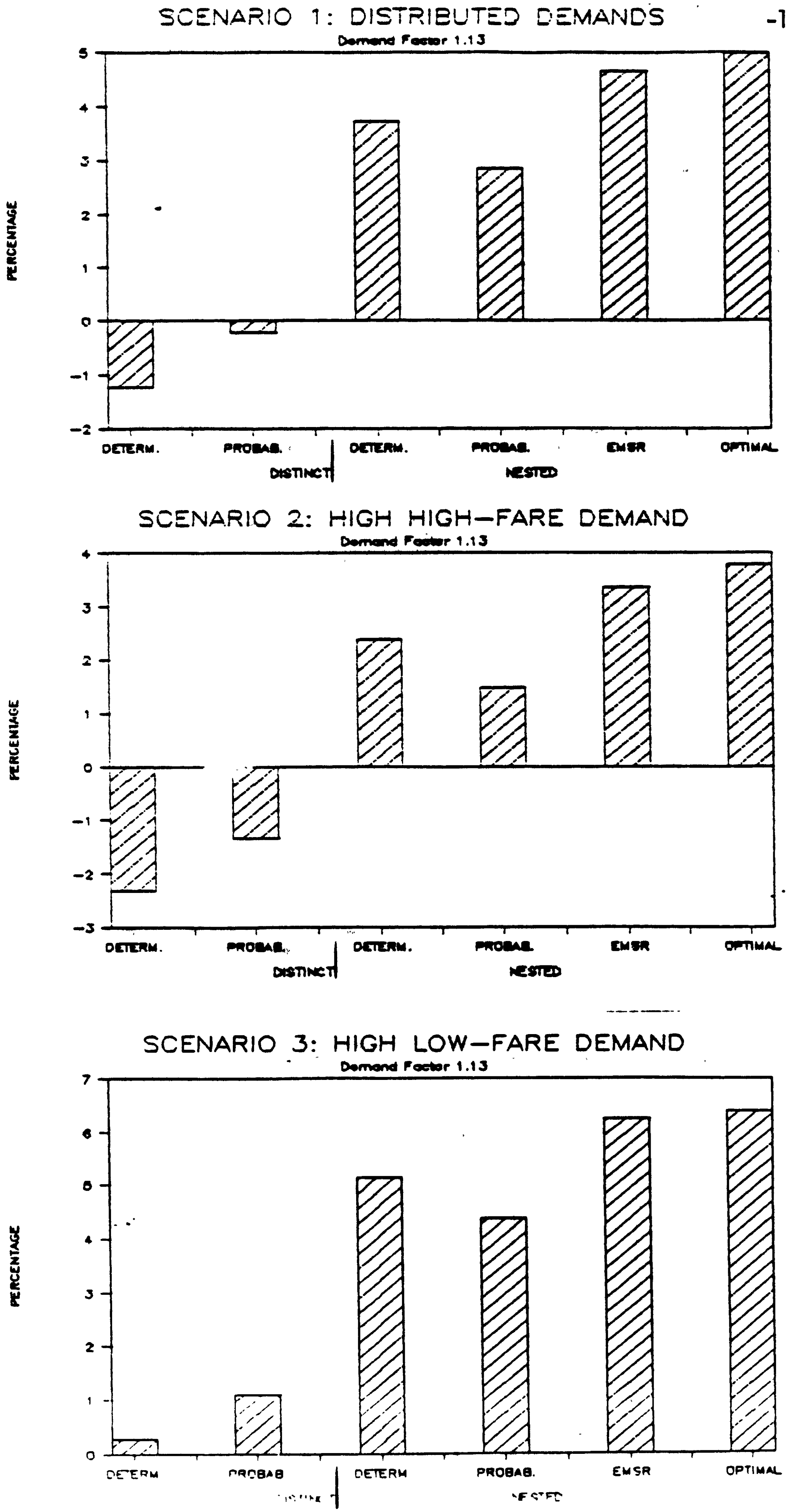
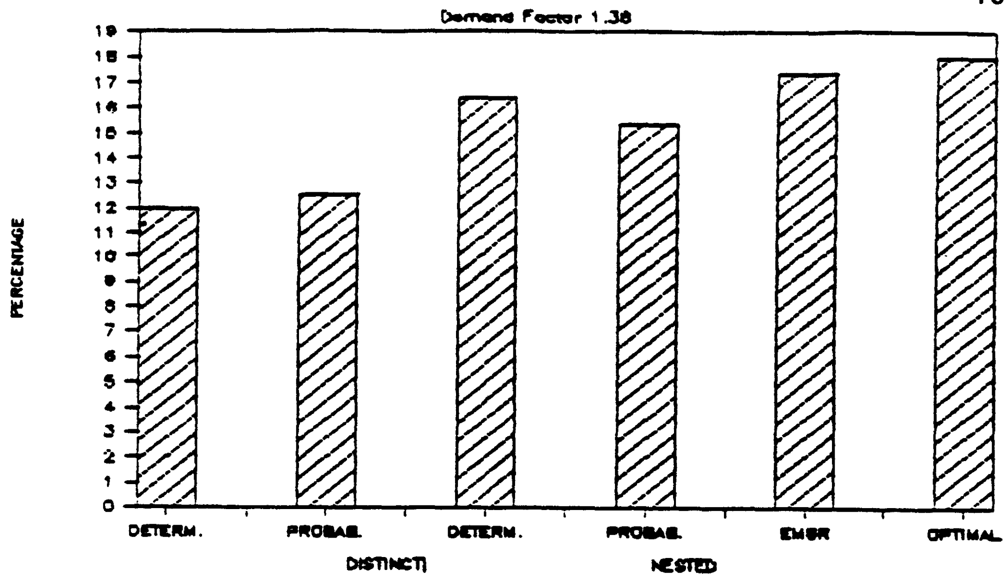
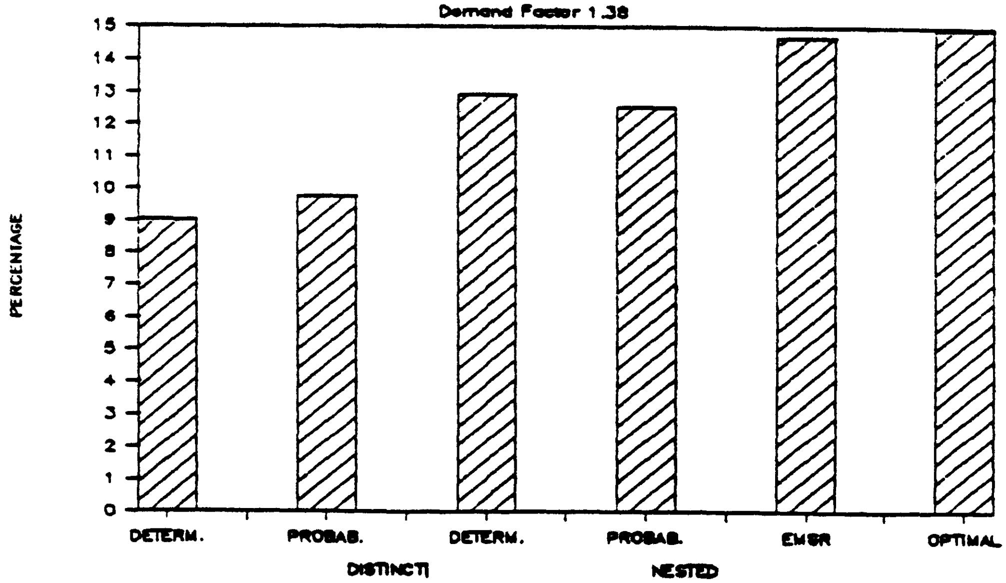

SCENARIO 1: DISTRIBUTED DEMANDS
Demand Factor 1.13
PERCENTAGE
5
4
3
2
1
0
-1
-2
DETERM.
PROBAB.
DETERM.
PROBAB.
EMSR
OPTIMAL
DISTINCT
NESTED
SCENARIO 2: HIGH HIGH-FARE DEMAND
Demand Factor 1.13
PERCENTAGE
4
3
2
1
0
-1
-2
-3
DETERM.
PROBAB.
DETERM.
PROBAB.
EMSR
OPTIMAL
DISTINCT
NESTED
SCENARIO 3: HIGH LOW-FARE DEMAND
Demand Factor 1.13
PERCENTAGE
7
6
5
4
3
2
1
0
DETERM
PROBAB
DETERM
PROBAB.
EMSR
OPTIMAL

FIGURE 7

SCENARIO 1: DISTRIBUTED DEMANDS



SCENARIO 2: HIGH HIGH—FARE DEMAND



SCENARIO 3: HIGH LOW—FARE DEMAND

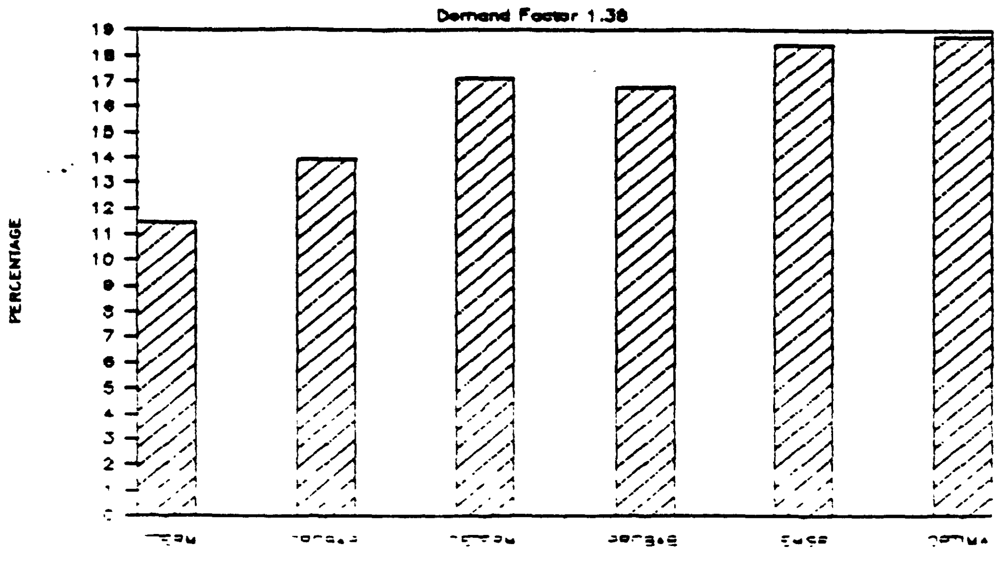


FIGURE 8

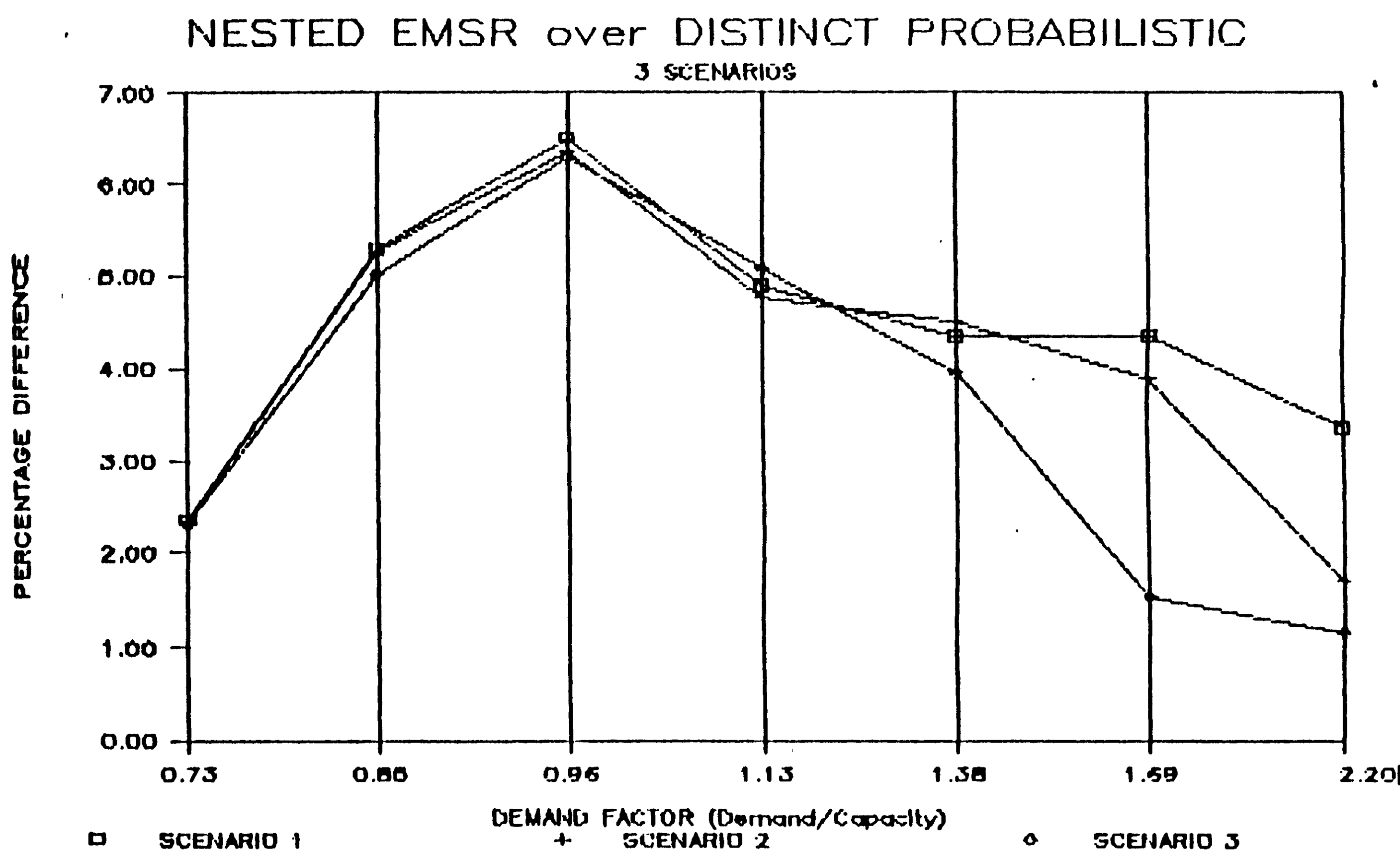

NESTED EMSR over DISTINCT PROBABILISTIC
3 SCENARIOS
PERCENTAGE DIFFERENCE
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
0.73
0.86
0.96
1.13
1.38
1.69
2.20
DEMAND FACTOR (Demand/Capacity)
SCENARIO 1
SCENARIO 2
SCENARIO 3

FIGURE 9

5. CONCLUSIONS

FLIGHT LEG CONTROL

- Important to match optimization method to booking class structure.

- Nested booking classes result in higher expected revenues than distinct classes.

- Probabilistic solutions outperform deterministic solutions

- Potential for negative revenue impacts occurs at low demand factors, especially with distinct classes.

- EMSR underperforms optimal nested solution by less than 0.5%, a small margin given:

  - uncertainty of input demand data

  - substantially greater processing time required to find optimal solution

References

Belobaba, Peter P., "Air Travel Demand and Airline Seat Inventory Management", MIT Doctoral Dissertation, May 1987.

Wollmer, Richard, "A Seat Management Model for a Single Leg Route When Lower Fare Classes Book First", Presentation to ORSA/TIMS Joint National Meeting, Denver, CO, October 25, 1988