FTL R81-8

IFA/1: AN INTERACTIVE

AIRLINE FLEET ASSIGNMENT MODEL

Herve Duchesne de Lamotte

Master's Thesis

May 1981

MIT

**DEPARTMENT
OF
AERONAUTICS
&
ASTRONAUTICS**

**FLIGHT TRANSPORTATION
LABORATORY
Cambridge, Mass. 02139**

IFA/1: AN INTERACTIVE

AIRLINE FLEET ASSIGNMENT MODEL

by

HERVE J-M DUCHESNE DE LAMOTTE

Diplome d'Ingenieur Civil de l'Aeronautique.
Ecole Nationale Superieure de l'Aeronautique
et de l'Espace
Toulouse France
(1979)

SUBMITTED TO THE DEPARTMENT OF
AERONAUTICS AND ASTRONAUTICS
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS OF THE
DEGREE OF

MASTER OF SCIENCE IN
AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1981

Signature of author_____
            Department of Aeronautics and Astronautics
                                      June 17, 1981

Certified by_____
                                 Robert W. Simpson
                                 Thesis Supervisor

Certified by_____
                                 Dennis F.X. Mathaisel
                                 Thesis Supervisor

Accepted by_____
                                 Harold Y. Wachman
            Chairman, Departmental Graduate Committee

1

To the memory of my Father

IFA/1: AN INTERACTIVE
AIRLINE FLEET ASSIGNMENT MODEL

by

HERVE J-M DUCHESNE DE LAMOTTE

ABSTRACT

This thesis investigates the Airline Fleet Assignment models
which have been designed during the past 10 years within MIT's
Flight Transportation Laboratory. Emphasis is placed on devel-
oping an interactive computer system, called IFA/1, which sim-
plifies the use of fleet assignment models and improves the
insertion or modification of necessary data.

The first section gives a general review of the mathematical
models used for vehicle planning in air transportation, with
special attention given to the Airline Fleet Assignment and
Fleet Planning problem. The techniques used to solve these
problems are discussed, and one model of particuliar interest,
the FA4 model, is introduced.

The second part describes how FA4 is used in practice and
points out some of its major deficiencies. An improvement is
proposed through the means of an interactive computer package
available at the Flight Transportation Laboratory. Future mod-
ifications to this interactive system are also discussed.

Finally, an alternative solution is suggested to one of the
theoretical problems which underlies the Fleet Assignment mod-
el: the "phantom frequency problem". This issue is discussed
and a new set of equations is proposed which improves the effi-
ciency of the model.

ACKNOWLEDGEMENTS

We should be able, one day, to achieve the synthesis of con-
sciousness. Above a certain level of complexity, which does
exist but is unknown, organic materials would become
conscious, and this very conciousness, higher in the scale of
evolution, would, in turn, become reflexive, aware of itself.

after Pierre Teilhard de Chardin.

1.0 CONTENTS

Contents   7

2.0 *INTRODUCTION*

2.1 *FLEET ASSIGNMENT - FLEET PLANNING*

Fleet assignment and Fleet planning are two major steps in an airline's decision-making process of determining the supply of air transportation services. They are strongly related since, usually, fleet planning needs fleet assignment as a prior operation. Let us see why.

1. Fleet assignment

Given a set of aircraft, stations to be served, a proposed route network, origin-destination demands for passenger/cargo, yields and costs, we want for a fixed length of time (day, week) and at a given date a three dimensional result: the "frequency vector". The frequency vector includes:

- which route should be flown,
- how many times, and
- by which aircraft type.

2. Fleet planning

Given the same data as before but over more than one time period (demand for the next 5 years or more); given the cost of selling, replacing or updating our aircraft and any financial constraints (e.g. cash availability, debt

limits, etc) imposed on the carrier; we want to know what will be our fleet requirements in the future:

- which aircraft should be retired,

- which new type and how many should be
  purchased/leased, and

- which airplane should be kept and retrofitted.

With these definitions in mind, one can see that fleet planning models will use fleet assignment techniques to decide which fleet best fits the proposed network. As we will see in the next chapter, the fleet planning problem has often been solved using a fleet assignment model. This technique, however, is not very efficient since the assignment has to be redone many times in order to cover a sufficiently long time period. The cost can be quite high; and the approach does not include fleet continuity from one period to the next, nor does it ensure an optimal solution over the entire planning horizon. How these two steps fit into the global airline supply of service scheme is described in Figure 1 on page 11.[1]

This figure shows that the "airline system" consists of two main sections:

---

[1]   This figure is inspired from one proposed by Professor D. F. X. Mathaisel from the Flight Transportation Laboratory during his lectures.

```
INPUTS          ┌─────────────┐
                │ Labor       │◄─────────────────────┐
                │ Fuel        │                      │
                │ Capital     │                      │
                │ Fleet       │                      │
                └─────────────┘                      │
                       │                             │
    ┌──────────────────┼─────────────────┐          │
    │            ┌──────▼──────────────────────────┐ │
    │            │ Transportation production function│ │
    │            │ performed by the following airline divisions:│
    │            │         - maintenance            │ │
    │            │         - flight operations      │ │
    │            │         - station operations     │ │
    │            │                                  │ │
    │            │ with:                            │ │
    │            │     Vehicle Routing and          │ │
    │            │   FLEET ASSIGNMENT MODELS        │ │
    │            └──────────────┬───────────────────┘ │
    │                           │                      │
INTERMEDIATE            ┌───────▼────────┐    ┌────────────────┐
OUTPUTS                 │ Network        │───►│ FLEET PLANNING │
                        │ Vehicle        │    └────────────────┘
                        │ Cost of service│             ▲
                        └───────┬────────┘             │
    │                           │           ┌──────────────────────┐
    │                           │           │ Fleet modification   │
    └───────────────────────────┤           │      costs,          │
                                │           │ Long range forecasts │
                                │           └──────────────────────┘
                        ┌───────▼─────────────────────────┐
                        │ Transportation scheduling function│
                        │ performed by the following divisions:│
                        │         - marketing              │
                        │         - scheduling             │
                        │                                  │
                        │ with:                            │
                        │                                  │
                        │   SCHEDULE GENERATORS            │
                        └───────┬──────────────────────────┘
                                │
FINAL OUTPUTS           ┌───────▼────────┐
                        │ Scheduled services│
                        │    prices      │
                        └───────┬────────┘
    └───────────────────────────┴── Consumer demand
```
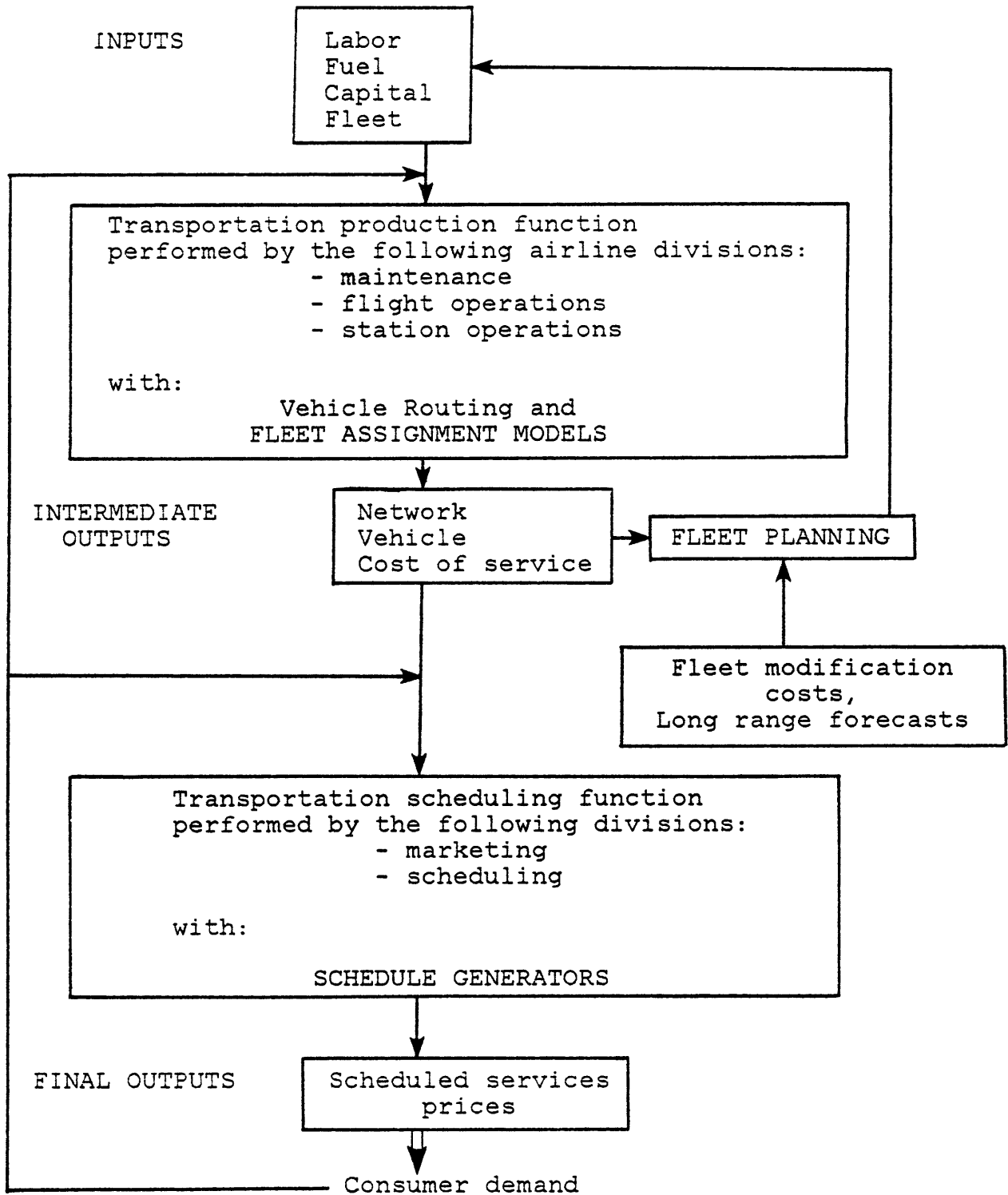
Figure 1. Airline supply of service: fleet assignment,
           fleet planning and scheduling are the main
           operations.

- "network selection" with its three dimensions: routes, vehicles, frequencies.
- "network coverage" or the scheduling of the vehicles on the chosen routes with the chosen frequencies.

It is a refinement procedure since there is no "time of day" consideration in the selection. The demand is considered on a daily (or weekly) basis. The scheduling process, at the other end, looks at the distribution of that demand throughout the day which means a more detailed analysis.

There are still many discussions about that double structure. Some people think that the assignment without the time of day information is not valid. They favor time-of-day scheduling where the output vector has an additional time dimension:

- routes flown,

- aircraft type used,

- frequency of service, and

- time of departure.

This fleet assignment with time-of-day scheduling optimization is, however, an extremely big and intricate problem which has never been satisfactorily solved. For the time being, the time-of-day issue is not included in our discussions since classical (non-time-of-day) fleet assignment formulations have given good results and are subject to other, more serious, criticism which will be raised in the next chapter.

2.2 *THE PRACTICE*

In practice, airlines perform their fleet planning "on the field" by trial and error.

When a new airline starts its new operations from ground zero, the aircraft used are those which were usually bought at low price, the routes served have been selected after some market research showing enough demand, and the frequencies are selected by trial in order to satisfy that demand. The schedule and the fares are established to compete as much as possible with other carriers.

Once the airline is firmly established in the market, future changes in frequencies and schedule are made from marketing research to remain as competitive and attractive as possible.

Fleet planning is also done by aircraft manufacturers who want to convince customer airlines that their products are needed. The airline provides the manufacturer with the number of seats required to fit their markets, the desired range, etc. The manufacturer will then assess which type of aircraft is best suited to the airline's needs on a long term basis. He will start drawings and then, with the caracteristics of the new design and computer models, will try to assess the total world-wide market for the aircraft.

As one can see, fleet assignment and fleet planning are not well automated within the airlines. The experience of the mar-

keting division as well as the "nose of a few old airline guys" were the principal tools used up to now.

"At 7:10 p.m., Air Florida flight 151 begins boarding passengers at Miami International Airport. Ed Acker is the last passenger to board. He takes a quick look around the cabin, counting empty seats. As usual, there aren't many. Acker settles his gangly, six-foot-four-inch frame into an aisle seat and pulls out an already well thumbed copy of the latest biweekly Official Airline Guide (...)
By the time Air Florida's blue and green Boeing 737 is airborne, Acker has filled two pages of yellow writing paper with notes and numbers. Within another hour or two, he will have found a couple of niches in a rival airline's schedule and devised a way to lure a few hundred more passengers a day onto his own planes.
Acker has been reading the Airline Guide - and nibbling away at the competition - for three years now "[2]

Computerized models for fleet planning were designed and used only by the manufacturers; but this is changing. Operations research is widespread and more and more carriers are willing to automate their "decision making". It is in line with this new trend that we decided to create a new interactive fleet assignment tool which could be used easily by airline people. It will be described in part two.

---

[2]   Ed Acker is chairman of Air Florida. See ref(36)

3.0 *THE MODELS USED*

In the past ten years, major air carriers started using computers to improve their operations. They created mathematical models or simply used those produced by more than 25 years of research in universities or corporate laboratories.
An overview of these models follows with emphasis on air transportation.

3.1 *VEHICLE ROUTING MODELS*

Vehicle routing models are a direct application of network flow theory. The problem to be solved is the routing of vehicles through demand points to pick-up or deliver goods at minimum cost. The most elementary (and most studied) example is the famous "Travelling Salesman Problem".
Numerous techniques have been used to solve this problem. They include simple heuristics, linear programming, cutting-plane methods and combinatorial optimization. Different linear programming formulations have evolved, where the emphasis is placed on the vehicle flow, commodity (traffic) flow, or on the network design problem. Practical applications, however, are difficult because of the problem size.[3] Decomposition techniques have been applied to decrease the size of the problem using Lagrangian Relaxation or Bender's technique. In

practice, however, these decomposition techniques are not used. Other tactical tricks, such as imposing a special network structure that attenuates the combinatorial nature of the problem, can be applied to simplify the Vehicle Routing Problem.

There are two classes of routing models in air transportation.

1. The aircraft routing models

   These models consider one aircraft at a time and follow it through the network in order to produce "rotations" (i.e., aircraft itineraries returning to their starting point). The objective here is to minimize the size of the fleet knowing operational constraints on the airplanes used. The solution is usually obtained using dynamic programming and linear programming.[4]

2. The fleet routing models

   These models are similar to the aircraft routing models. Instead of looking at one aircraft they deal with the whole fleet at once. As Professor Simpson from MIT's Flight Transportation Laboratory says [5]

---

[3] As reported by Professor Mathaisel a commodity flow formulation can lead to 20,300 constraints, 10,000 discrete variables and 1 million continuous variables.

[4] See ref (1),(12) and (14)

[5] See ref (26), and for more details on fleet routing ref (15),(25)

"All of the fleet routing models can be posed as network flow problems on a network where the circulation flows must be integer."

This deduction is why four fleet routing models proposed by the laboratory can be solved using an Out of Kilter algorithm. They are:

- FR1 - We want a minimum fleet for a fixed schedule.

- FR2 - What set of services should be flown in order to maximize the income, knowing costs and fares, given a schedule of non-stop services?

- FR3 - Same as FR2, but with a fixed fleet size.

- FR4 - Here we still have the same problem, but multi-stop flights are possible by allowing non-stop services to be linked together.

Even if time-of-day information is used, these models are not very precise and cannot be applied easily since only one type of aircraft is allowed and no distinction is made between individual vehicles.

3.2 ASSIGNMENT MODELS

This class of models have been given a lot of attention since

---

[6] As early as 1955, G. B. Dantzig himself published an application of linear programming which was a fleet assignment problem. See ref (3),(4)

they can be easily expressed and solved with linear programming techniques.[6] Four types of models have been designed for air transportation.

1. Aircraft assignment models

    An aircraft type is assigned to a given set of routes. The objective is to find the optimal allocation of aircraft to this set of routes, subject to various operational constraints.

2. Frequency determination models

    We want to find the optimal flight frequencies on a given route network. These frequencies usually depend on demand and capacity which are also given in the model. Integer linear programming is used to solve this problem. The fleet assignment models designed at MIT's Flight Transportation Laboratory also include the aircraft type in their results and the demand is itself a function of the frequency. These models will be described in more detail in the next chapter. It must be pointed out that, for the time being, because of their size, these models are studied with non-integer linear programming techniques.

3.3 *SCHEDULE CONSTRUCTION MODELS*

These models are also named "Time of Departure Determination

Models". For a given set of flights and the distribution of demand throughout the day, find the best departure time for the aircraft. Heuristics have often been used to solve this problem which is the basis of airline scheduling.[7]

3.4 PASSENGER ALLOCATION MODELS

Here the formulation is similar to that of the Routing, Assignment, and Frequency determination models but traffic flow is added. It can be either passenger or cargo flow through the network. The problem can be interpreted as a multiple commodity network flow problem, where each commodity represents passengers (or cargo) between a certain origin and destination.[8]

3.5 OTHER MODELS

Routing and Assignment problems are certainly the most studied, and the Flight Transportation Laboratory produced computer programs of both models. This is why emphasis was placed on them, but there are many other models used by airlines, manufacturers and researchers. Every year, new schedul-

---

[7]   See ref (18)

[8]   See ref (6)

ing algorithms are discovered, which are better suited to very specific problems, and the list of new heuristics is endless.

Certain areas, however, are becoming more popular: fleet planning is one of them. The reasons are quite clear: aircraft are becoming more and more expensive and airlines are becoming more capital intensive. In order to remain competitive, be adapted to their markets, and be financially sound, airlines have to order the proper model of aircraft. "Proper" means that the aircraft must best fit their needs, not only in the near future (2 to 3 years), but much farther away (up to 15 years), since aircraft delivery times are growing longer and longer. On the manufacturer's side, the decision to make a new aircraft type should be carefully studied as well, demonstrating in particular that there is definitely a market big enough.

At the macroscopic level of planning, manufacturers need to know which kind of aircraft will be required on a worlwide basis. They generally use heuristic techniques such as the "ASM Gap" (or Aircraft Productivity Technique), which consists of 5 main steps:

- Forecast the world passenger and cargo traffic.
- Under certain load-factor assumptions, convert traffic into needed-capacity forecasts.

---

[9] The productivity of an aircraft is the product of its speed, its utilization, and its capacity.

- Project out the available capacity (no new aircraft being produced) from the actual fleet less subsequent retirements, under various service life scenarios and productivity.[9]
- Calculate the gap between needed and available capacity (which will be filled by new aircraft).
- Taking into account aircraft mix, availability and costs derive the needed number of aircraft with associated revenues and costs.

At the microscopic level of planning, airlines want to know which aircraft will be best suited to their network and type of operations. They want a two-step study:

- traffic forecast for the coming years.
- fleet assignment for each year with the economics of buying and selling aircraft included.

Until recently, the technique used was roughly to run a fleet assignment model for each year, using the intermediate results to update the fleet and the network.

However, using the concept of cell theory, where route network information is aggregated,[10] Professor D. F. X. Mathaisel has designed a new fleet assignment model which is already in use today by airlines and manufacturers.[11]

Finally, in between these two models, there are a lot of dif-

ferent variations used by the manufacturers and the carriers.

They range from MACRO models at the world scale to MICRO prob-

lems at the route level.

---

[10] For example, instead of dealing with individual routes, it
is possible to group elements of the network into a "cell"
whose attributes are distance (200 to 300 miles), demand
(1,000 to 1,500 passengers per week) and service (15 to 28
flights per week). See ref (20)

[11] See ref (10),(13),(17),(19),(22)

4.0 *THE FA. MODELS*

Over the past ten years MIT's Flight Transportation Laboratory has studied and designed a complete series of fleet assignment models (see ref (27)). They were called FA models and range from FA1 to FA7. Emphasis will be made here on FA4 which is the focus of this research.

4.1 *THE FA. SERIE*

1. FA1

    Find the minimum cost assignment of aircraft to routes for a given time period.

    - Demand is fixed

    - Routes are all non-stop

    The result is a least cost frequency pattern.

2. FA2

    It is the extension of FA1 to multi-stop routes.

3. FA3

    This model is again an extension of FA1 where the demand is not fixed. The passenger demand is now a function of frequency and is entered into the formulation with "demand vs frequency" curves. The results include also the passengers carried in each market.

4. FA4

FA4 is an extension of FA2 to the variable demand:

- Demand is function of frequency

- Routes are multi-stop.

The equations of this model are presented afterwards. (See ref (28))

5. FA5

It extends FA4 to connecting services.

There are passenger paths, different from aircraft routes, which only use parts of these routes. (See ref (29))

6. FA6

This model is an FA4 formulation which includes a solution to the phantom frequency problem. This problem is of particular interest and is discussed in part 3 of this thesis. (See ref (11))

7. FA7

FA7 is another extension of FA4 to mixed demand. Passengers are separated into business travellers (sensitive to time) and pleasure travellers (sensitive to fares).

Pleasure passengers can use either discount fare services or regular scheduled services. Business can only fly with scheduled services. Each class, however, influence each other since, for example, if pleasure passengers start using regular flights, the resulting increase in frequency will stimulate business demand. (See ref (31))

4.2 FA4

The following equations explain in detail how the fleet assignment problem has been modelled with linear equations in FA4.[12] They are presented here in a Linear Programming formulation.

The notation used is explained in Appendix B. The reader will also find in part three of this thesis more details on these equations and on the "frequency vs demand" curves.

1. Objective function.

   Maximize the airline profit: revenue less expenses.

$$\text{Maximize} \ \sum_m \sum_{\underline{R}_m} Y^m \ T^m_r \ - \ \sum_r \sum_v VAC_{vr} \ n_{vr} \ - \ VIC_a \qquad (1)$$

2. Constraints.

   There is a maximum allowable load-factor on all route links.

$$\sum_v MAX \ LF^l_v \ S_v \ n_{vr} \ - \ \sum_m \sum_{\underline{M}^l_r} T^m_r \ \geq \ 0 \qquad (2)$$

   for all links l of routes r.

---

[12]   See ref (32),(33),(34),(37)

The total traffic carried on market m is a function of the total frequency.

$$\sum_i SL_i^m\, n^{mi} - \sum_{\underline{R}_m} T_r^m \geq 0 \qquad (3)$$

for all markets m.

The total frequency is a weighted sum of multi-stop frequencies.

$$\sum_s \sum_v K_{vs}^m \sum_{\underline{R}_s^m} n_{vr} - \sum_i n^{mi} = 0 \qquad (4)$$

for all markets m.

Fleet availability.

$$\sum_r \sum_v BT_{vr}\, n_{vr} \leq MAX\ U_v\, A_v \qquad (5)$$

for all aircraft types v.

Minimum level of service with flights up to M stops.

$$\sum_{s=1,M} \sum_{\underline{R}_s^m} \sum_v n_{vr} \geq MIN\ n^m \qquad (6)$$

for all markets m.

Maximum daily number of departures at a station.

$$\sum_v \sum_h \sum_{R_{hj}} n_{vr} \leq \text{MAX } N^j \tag{7}$$

for all stations j.

3. Bounds

The frequency on each segment of the demand curve is bounded.

$$0 \leq n^{mi} \leq n_{mi} - n_{m(i-1)} \tag{8}$$

for all markets m and all segments i.

5.0 *INTERACTIVE PROGRAMS - RULES*

5.1 *INTERACTIVE EXCHANGES*

It is difficult to give a precise definition of *INTERACTIVE COMPUTER MODELS*. This is mainly due to the wide variety of interactive processes which can be created. Examples of such processes will follow. A general characteristic however, can be outlined: these processes are very close (or at least, as close as possible!) to a *conversation*[13] between two human beings. One of them is the *master,* (usually extremely lazy!) the other one, the *slave* who never thinks but executes...

The use of the word *CONVERSATION* may not be, in fact, the best. *EXCHANGE* is more appropriate, since words and sentences are not necessarily involved. Means of communication between people and computers are numerous and their variety is related to the interfaces[14] available on the market.

---

[13] Interactive dialogues are often a part of a "conversational" system. IBM's CMS stands for "Conversational Monitor System".

[14] "Interface" is a general word representing all the devices used to exchange information between two "units" which understand the same data, but read or write them in a different way.

One can think about:

graphics

This medium is the main focus of research today. The interface is a screen which displays any kind of picture. The operator points out a part of the graphic in order to start computer processing.

conversation

The computer and the operator ask themselves questions and answer them with common words and sentences. The computer game "Adventure" is a good example.[15] Another very realistic game is the computerized "psychoanalist" designed by artificial intelligence researchers.[16]

There is still a lot of research going on, since as in an ordinary conversation, answers or questions can be of any kind and use any possible word. The main difficulty is the interpretation to the computer of complete sentences. The computer needs to recognize the important words or "keywords", which

---

[15] "Adventure" is an interactive game which is similar to the well known "Dungeons and Dragons".

[16] A computer asks questions to a "patient" as a psychoanalyst would do. The conversation which takes place is extremely realistic.

involves very complex language syntax and grammar analysis.

coded conversation This medium is the most commonly used interactive process which also works with questions and answers between the computer and the user. The language however, consists of coded instructions. The freedom of syntax is very narrow and a "users manual" is needed. Coded conversation is how the interactive FA4 preprocessor has been designed.

The diversification of interfaces and higher speed of computers has greatly improved *interactive exchanges*. For this reason the science of man-computer communications is fairly new and there are still no established rules in dialogue engineering[17] . The literature is very poor on this subject. No systematization of the technique has been done: neither in Martin's _Design of Man Computer Dialogues_ (the first look at the topic in 1973); nor, later, in B. R. Gaines' work. Gaines, however, made recently a first attempt in his conference _Programming Interactive Dialogue_ . This conference as well as

---

[17] We are in the position today in programing man-computer interaction that we were with hardware design thirty years ago and software design ten years ago.

[18] See ref (7),(8),(9),(16)

experience and common sense inspired the following set of basic
rules.[18]

5.2 ELEMENTARY RULES FOR INTERACTIVE DIALOGUE

5.2.1 DESIGN OF THE SYSTEM

Rule 1.   The activity (e.g. a fleet assignment or any design
problem) should be modeled to take into account current
procedures in order to *provide a familiar environment*.

Rule 2.   The design should be *updated continuously*. Future ses-
sions and accumulated experience will be used as well
as the system capabilities themselves.

Rule 3.   Activities, especially errors on the user's side,
should be *recorded* for future system-design modifica-
tion.

5.2.2 USER-SYSTEM RELATIONSHIP.

Rule 4.   The user should not be passive and controlled by the
computer. *He will command the system*.

Rule 5.   *The user should dominate the system*, which means a
non-specialized design since:

  •   if the computer dominates, it should provide
       enough information to do so and by that means,

guide, direct or simply model the user to it's own behavior.

- if the user dominates, the model must be clear, simple to understand and should leave *total freedom of decision* concerning the task which should be executed next.

Rule 6. As a part of the clarity of the system, each *given action* on the user's side should lead to a *given response* from the computer. There should not be any "black holes" where the user does not even know or remember which process he started.

Rule 7. The system environment should not be *complicated*:

- The beginner should be helped as much as possible by extensive and explicit messages.
- The experienced user must be able to "fly" through the procedures as quickly as he wants.

Rule 8. Simplicity. The commands must be easy to write and as logic as possible. Complicated syntax and options should be avoided.

5.2.3 *THE MASTER IS LAZY: MINIMIZING THE MENTAL WORKLOAD*

Rule 9. The system should be *uniformly organized*. All the commands and processes must look alike.

Rule 10. There should be a *gradual Help-Command*.

- The first call gives a very short but complete answer to remind the user.

- Another call, immediately following the first one, gives more information to show the user.

- Further calls would release enough details to teach the user.

Rule 11.   The position of the user in the system should be *reminded* all the time by various messages or special prompts specific to each step or level. The user should never be lost.

Rule 12. There should be a command to *abort any activity* and come back to a previous step or a higher level.

Rule 13. The user should be able to come back and *correct* his mistakes in the middle of a procedure, without having to go through it all before attempting any correction.

Rule 14. The user should be able to see the pertinent data stored in memory before correcting any records.

5.3 SUMMARY

The above list of rules is certainly not complete, but provides enough guidelines to shape any interactive system. Of course, further extensions can and will be made, especially since the previous set mostly applies to exchanges through

"hardcopy terminals" where the conversation goes on "line by line" and never "page by page".

6.0 THE FLEET ASSIGNMENT PROCEDURE IN FA-4

In order to comply with rule 1 (the environment should be familiar), the first step of the interactive design is to study the fleet assignment *procedure,* to model the system around it, and make the new environment look somewhat familiar.

6.1 THE OLD PROCEDURE.

As we have seen in part one, the fleet assignment model consists of simultaneous linear equations. All the FA. models we discussed were expressed in term of linear programming, which already means 3 main steps:

*   setup the equations

*   run the optimization

*   report the results

Let us now expand these steps:

1. Many steps are needed in order to write the equations in linear program form. We have to:
    *   insert the data
    *   verify the data
    *   translate the data into simplex matrix form.

2. to solve the linear program means to have access to a com-
   mercially available L.P. package.

3. to present the results means to read them and translate them
   into a form which is understandable to the user.

 Therefore, the FA-4 model was designed with the structure
described in Figure 2.

 If we go one step further, the structure becomes a more
divided and outlines the main sections of the FA4 program.

1. The preprocessor is organized around the data which are
   themselves divided into groups:

   system data :   these include general economic and system
                          data.

   aircraft data :   technical and economic information about
                          the aircraft available.

   airport data :   name of the cities the airline wants to
                          serve with possible constraints.

   city-pair data :   description of all the markets we might
                          serve: distance, fare, demand information.

   route data :   definition of the route network.

 The order in which these groups are listed is the same as
the input order in FA-4 and goes from general to more spe-
cific data. Actually, each subsequent set of data to be

```
              Input Data

                  │
                  ▼
        ┌──────────────────────┐
        │     Preprocessor     │
        └──────────────────────┘
                  │
                  ▼
        ┌──────────────────────┐
        │    Linear Program    │
        └──────────────────────┘
                  │
                  ▼
        ┌──────────────────────┐
        │    Postprocessor     │
        └──────────────────────┘
                  │
                  ▼

           Readable Output
```
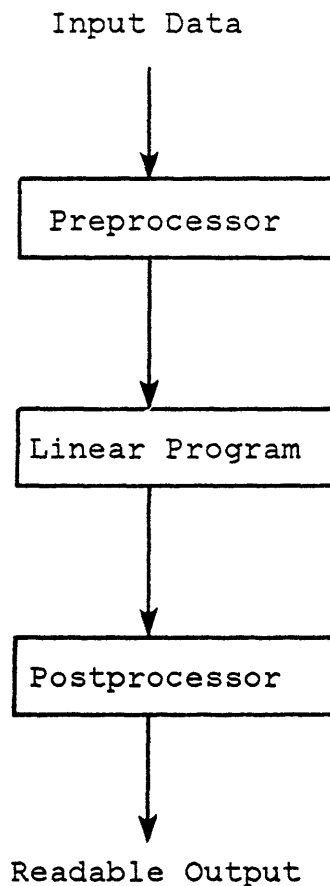
Figure 2.   FA4 traditional stucture: It shows how FA4 was
            previously organized.

input often needs information from the previous groups of

data in order to be processed. For example, the station

names in the route data set can only be checked if the city

names are available.

2. One of the data-sets itself can be indirectly created: the

city-pairs. The *level of demand* information is provided by

the *LEVEL OF SERVICE VS. DEMAND CURVES*, described in more

detail in chapter 3. The user can either provide curve

points directly or have the curve computed from historical data. This is done by a program called *THE DEMAND CURVE GENERATOR*.

3. The linear program requires a specialized input format. The main task of the preprocessor is then, after various checks and some transformations of the input data, to write the equations in the special form and to send this information into a new file which will be used as input to the L.P.

4. The L.P. in turn, provides the results of the optimization in a standard mathematical programming format. This output must be processed again in order to be understood by the user. This is the task of the postprocessor. The information provided at this step have been divided into:

  • *route data*: aircraft used, frequencies of flights, detail of passengers carried.

  • *segment data*: same as the route data but broken down into flight segments.

  • *city-pair data*: here we have the Origin and Destination (O & D) results. The link between both cities can be a multi-stop flight.

  • general *economic results* for the system.

  • system *activity and utilization*.

Figure 3 on page 40 represents the detailed FA4 structure.

```
┌─────────────────────────────────┐                    ┌──────────────────┐
│ Historical Passenger Traffic    │───────────────────▶│   City Pair      │
│ Historical Levels of Service    │                    │ Demand Curve     │
└─────────────────────────────────┘                    │   Generator      │
                                            ┌──────────▶│                  │
┌─────────────────────────┐                 │          └──────────────────┘
│ Demand Formulation      │─────────────────┘                   │
└─────────────────────────┘                                     │
                                                                ▼
┌──────────┐   ┌──────────┐   ┌──────────┐   ┌──────────┐   ┌──────────┐
│ Economic │   │ Aircraft │   │ Airport  │   │ Demand   │   │ Route    │
│ Data     │   │ Data     │   │ Data     │   │ Data     │   │ Data     │
└──────────┘   └──────────┘   └──────────┘   └──────────┘   └──────────┘
       ╲           ╲              │              ╱              ╱
        ╲           ╲             │             ╱              ╱
         ╲           ╲            ▼            ╱              ╱
┌─────────────────┐          ┌──────────────────┐
│ Policy Options  │─────────▶│   Preprocessor   │
└─────────────────┘          └──────────────────┘
                                      │
                                      ▼
                        ┌──────────────────────────────┐
                        │ Mathematical Formulation     │
                        │ Linear Program               │
                        └──────────────────────────────┘
                                      │       Solved by
                                      ▼
                        ┌──────────────────────┐
                        │ MIT Package          │
                        │ "SESAME"             │
                        └──────────────────────┘
                                      │
                                      ▼
                        ┌──────────────────────┐
                        │   Postprocessor      │
                        └──────────────────────┘
                                      │
                                      ▼
                        ┌──────────────────────┐
                        │   Output Data        │
                        └──────────────────────┘
```

By Market   System-wide

```
┌─────────────────────────────┐          ┌──────────────────────────┐
│ .Flights                    │◀─────────│ .Profit or Loss          │
│ .Aircraft                   │          │ .Revenue                 │
│ .Passengers carried         │          │ .Fleet                   │
└─────────────────────────────┘          │ .Passengers              │
                                         │ .Airport                 │
                                         │    Activities            │
                                         └──────────────────────────┘
```
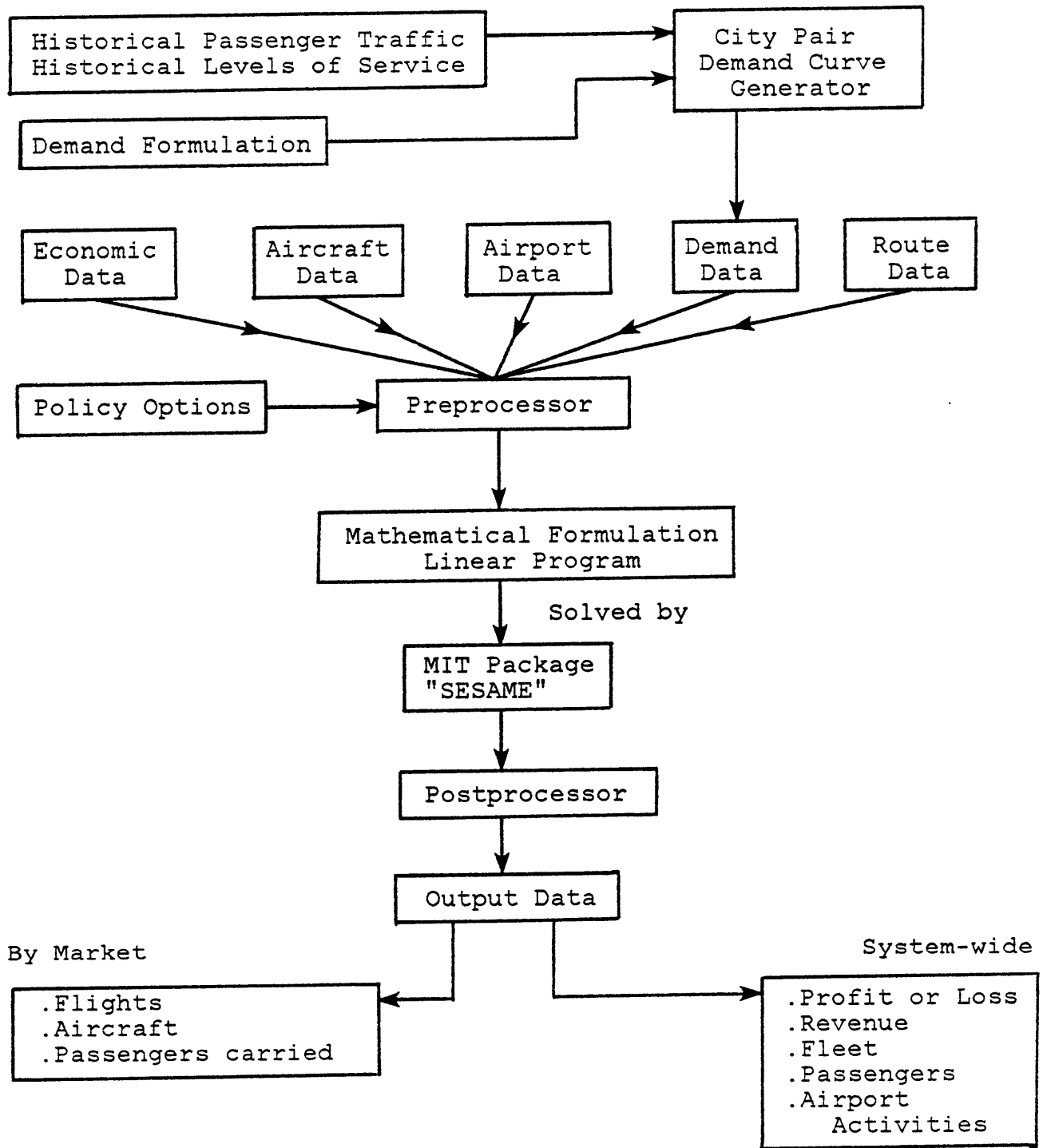
Figure 3. Detailed FA4 structure: This figure details the
           input and output data.

6.2 _THE WEAK POINTS_

1. Input data.

    Two examples of both input and the output are provided in Appendices C and D. A quick look at the input outlines what was clearly the main problem with FA4. All the input data had to be entered on a long formatted file. This process is subject to many errors.

    To modify the data was also difficult. One had to know exactly which figures to change and sometimes, especially with more than 500 city-pairs, the search for a given element could be very long. Of course, in the recent runs all this information was stored on files accessible with an editor[19].

    Therefore, the use of FA4 was restricted to people with a certain knowledge of computer systems; and even for these individuals the input of data was cumbersome.

2. Output data.

    The results coming out from the postprocessor were either printed on listings or directed to a file. In either case, however, the display of information would look like the sample provided in Appendix D . Beside weaknesses in the

_____

[19] The use of a system editor means, of course, that only a user familiar with the operations of that editor can work properly with FA4.

format itself, this type of output is not suited at all for further processing. For example, if a user wants to try five different scenarios and plot the trends of certain results, the output information needs to be stored in another way. It is best that the output be:

- more *compact*, and
- more *accessible* by a program for comparison, classification, plotting, etc.

6.3 SUMMARY

The first improvement needed for FA4 is the way the input of data is handled. This is why the interactive FA4 has been designed. Initially it will allow the user to provide information for the optimization much more easily. Moreover, after a run it will be possible to modify the "old data" with special commands.

The improvement of the output is a suggestion for future work since it is a much longer task. The postprocessor also needs to be completely redone in order to achieve the necessary flexibility.

7.0 *THE INTERACTIVE SYSTEM*

The preliminary question is, of course, which computer lan-
guage should we use? FA4 was written in FORTRAN, and at first
there was a strong tendency to remain consistent and retain
that language. However, due to the benefits of *list processing*,
and the enormous enthusiasm of Professor A. L. ELIAS from MIT's
Flight Transportation Laboratory, it was decided to use PL/1
for what will be called the *pre-preprocessor*.

7.1 *STRUCTURE OF THE INTERACTIVE FA4*

1.   *The general structure.*

Because of the weak points listed in the previous chapter,
the first thing needed was a complete change of the way
input data should be entered. It was decided, then, to keep
the   original   structure   of   FA4
(preprocessor-postprocessor).   The   input   data   file
stucture would remain unchanged with its five sections and
a *pre-preprocessor* would be created in order to build or modi-
fy these files. The interactive FA4 has, then, the
structure described in Figure 4 on page 44.

2.   *The pre-preprocessor.*

Some experience had been gained in the past on creating or
modifying FA4 data decks, but the real definition of the
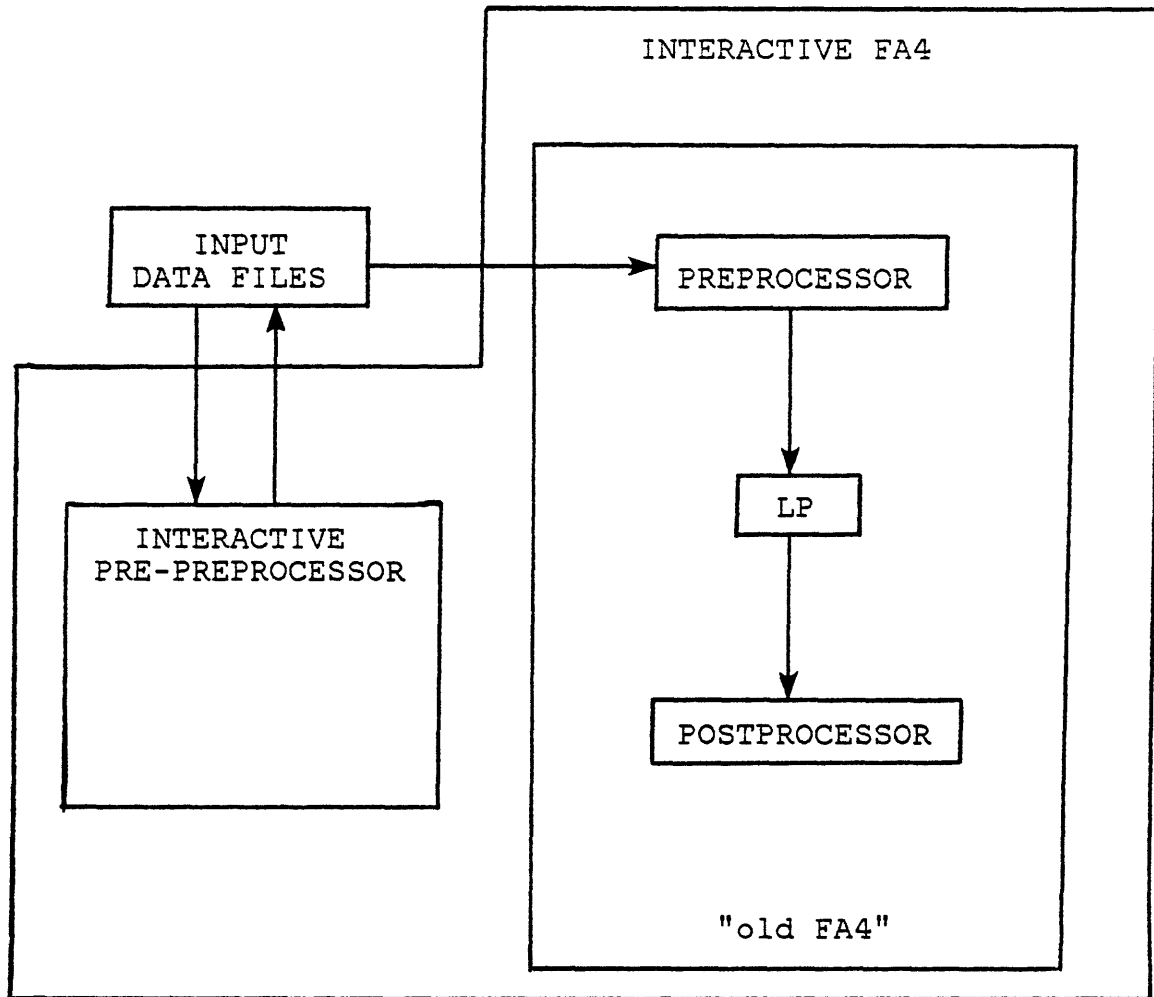
Figure 4.   Interactive FA4:   general organization of the
            system.

commands could only come after using an interactive system itself. In fact, the commands available in the actual version of the pre-preprocessor were implemented after discussions with Professor R. W. SIMPSON, Professor A. L. ELIAS and Professor D. F. X. MATHAISEL as well as an extensive personal use of FA4 on a realistic problem for Hughes Airwest.

The new stucture described herein is therefore better suited to the user's needs. It has two advantages:

a. to keep the traditional "working" FA4 unchanged, and

b. to be ready for further modifications since the interactive module is separate.

Figure 5 on page 46 shows how the pre-preprocessor is organized. It was designed with the ideas of *creation* and *modification* in mind. We shall now examine that structure.

Since the data are stored on disk files, they cannot be modified directly. All the information has to be transfered into the computer working memory, where it can be manipulated. For easier processing the data that has been read is organized into *"PL/1 LISTS"*. This is why the following two very important steps appear in Figure 5 on page 46.

a.   read the data from disk files into computer working memory

   CREATE LISTS FROM FILES

```
                    ┌─────────────────────┐
                    │  New or old files ? │
                    └─────────────────────┘
              NEW       ╱             ╲        OLD

    ┌──────────────────────┐      ┌──────────────────────┐
    │       Create         │      │       Create         │◄┄┐
    │  new data lists      │      │   data lists         │  ┊
    │      from            │      │      from            │  ┊
    │    keyboard          │      │  old disk files      │  ┊
    │      or              │      └──────────────────────┘  ┊
    │ tapes, cards...      │                 │              ┊
    └──────────────────────┘      ┌──────────────────────┐  ┊
              │                   │   Add new data       │  ┊
    ┌──────────────────┐          │      or              │  ┊
    │   Verify         │          │ modify old data,     │  ┊
    │   the data       │          │   erase them         │  ┊
    └──────────────────┘          └──────────────────────┘  ┊
                                             │              ┊
                                  ┌──────────────────┐      ┊
                                  │   Verify         │      ┊
                                  │   the data       │      ┊
                                  └──────────────────┘      ┊
                                                            ┊
    ┌──────────────────────────┐         ┌──────────────┐  ┊
    │ (Re)create disk files    │┄ ┄ ┄ ►  │  DATA        │  ┊
    │    from the lists        │         │  files       │──┘
    └──────────────────────────┘         └──────────────┘
                                                │
                                         ┌──────────────────┐
                                         │  Preprocessor    │
                                         └──────────────────┘
                                                │
                                            ┌────────┐
                                            │  LP    │
                                            └────────┘
                                                │
                                         ┌──────────────────┐
                                         │  Postprocessor   │
                                         └──────────────────┘
```
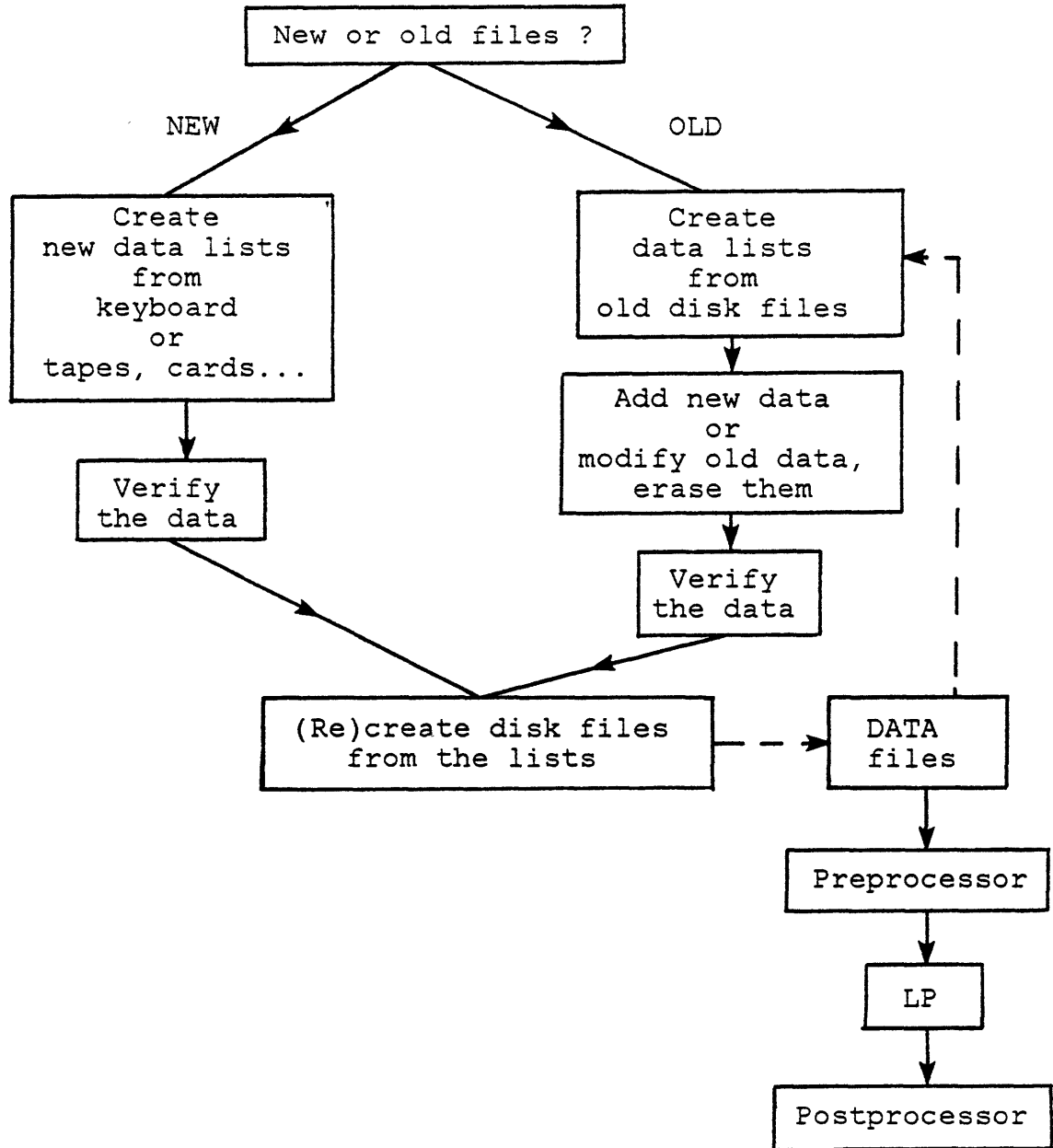
Figure 5. Structure of the data processing: it shows the
transfers made between disk files and lists.

b. send back the processed data into the files
    (RE)CREATE FILES FROM LISTS

The next figure (Figure 6 on page 48) shows the creation
of new data files.

In these boxes a new "entity" appears called the *MAIN
READER*. It is due to the internal organization of the com-
mands which is easily obtained with PL/1: namely there are
several levels of commands. In our case three levels are
implemented. [20] Here also, lists are used in lieu of tables
in order to give a greater flexibility and efficiency. If
the data entered at the first prompt are not correct, there
is a possibility to change them within the *"prompt module"*.
But if the decision is made to modify these data later in
the session, the creation of a list makes that
information-deck look as if it were an old one, which means
it can be accessed with the other section of the
pre-preprocessor dealing with modifications rather than
creation.

The structure of this section is shown in Figure 7 on page
49.

The only difference here is level two. It is a sublevel of
"modification-commands" adapted to each type of file.

___

[20] Figure 8 on page 51 shows how the three levels are related
to each other.

LEVEL 1

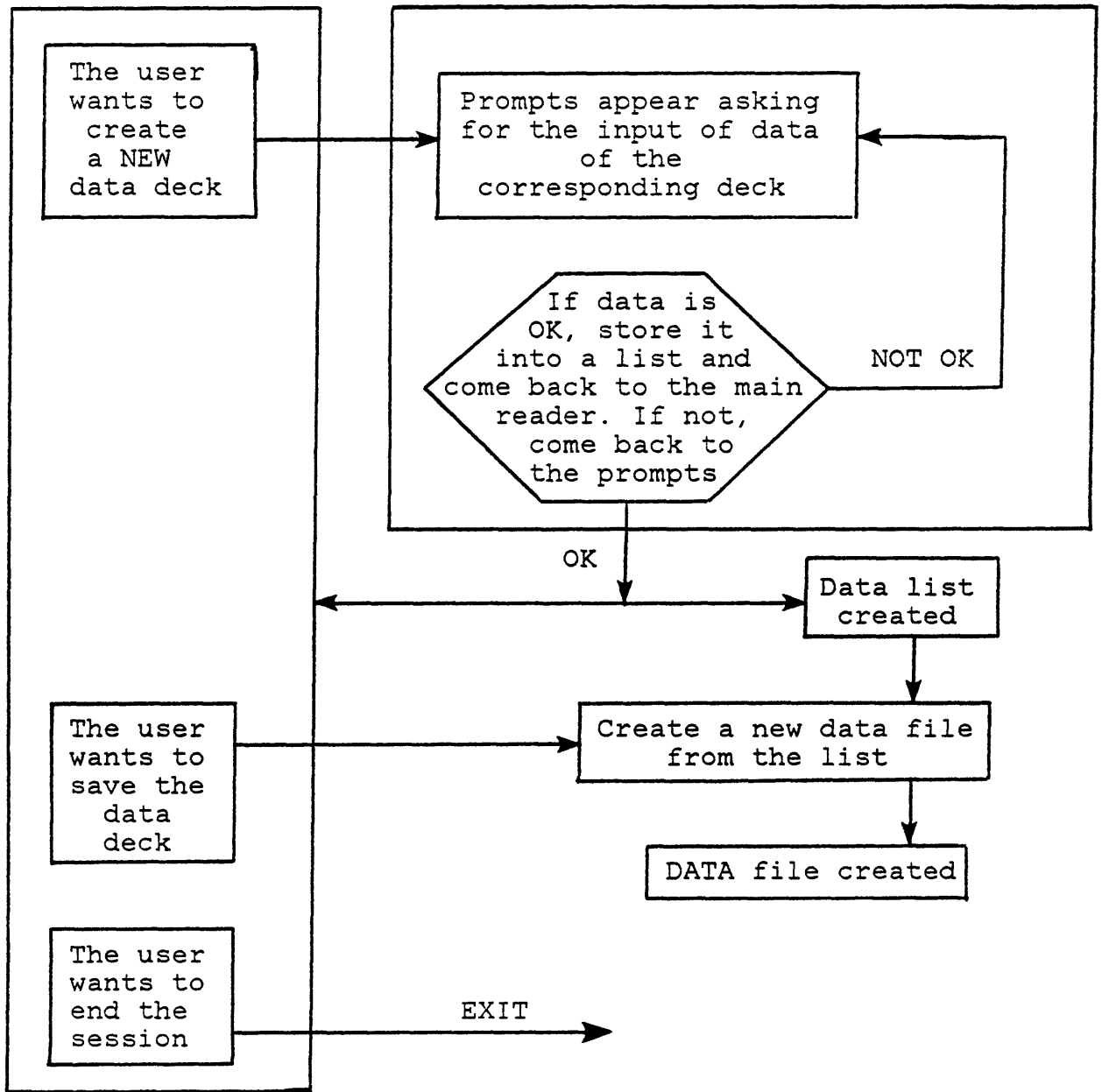Main reader   Prompt module

```
┌──────────────────┐  ┌──────────────────────────────────────┐
│ ┌──────────────┐ │  │  ┌────────────────────────┐           │
│ │ The user     │ │  │  │ Prompts appear asking  │◀──────┐   │
│ │ wants to     │ │  │  │ for the input of data  │       │   │
│ │ create       │─┼──┼─▶│        of the          │       │   │
│ │ a NEW        │ │  │  │   corresponding deck   │       │   │
│ │ data deck    │ │  │  └────────────────────────┘       │   │
│ └──────────────┘ │  │            ╱╲                      │   │
│                  │  │          ╱    ╲                    │   │
│                  │  │        ╱ If data is ╲              │   │
│                  │  │      ╱  OK, store it  ╲            │   │
│                  │  │    ╱  into a list and   ╲  NOT OK  │   │
│                  │  │   ╱ come back to the main╲─────────┘   │
│                  │  │    ╲ reader. If not,    ╱              │
│                  │  │      ╲ come back to   ╱                │
│                  │  │        ╲the prompts ╱                  │
│                  │  │          ╲        ╱                    │
│                  │  └────────────╲─OK─╱─────────────────────┘
│                  │                 │
│                  │◀────────────────┼──────────────▶┌──────────────┐
│                  │                 │               │ Data list    │
│                  │                 ▼               │ created      │
│                  │                                 └──────────────┘
│ ┌──────────────┐ │                                        │
│ │ The user     │ │                                        ▼
│ │ wants to     │ │              ┌────────────────────────────┐
│ │ save the     │─┼─────────────▶│ Create a new data file     │
│ │ data         │ │              │      from the list         │
│ │ deck         │ │              └────────────────────────────┘
│ └──────────────┘ │                              │
│                  │                              ▼
│ ┌──────────────┐ │                  ┌──────────────────────┐
│ │ The user     │ │                  │ DATA file created    │
│ │ wants to     │ │                  └──────────────────────┘
│ │ end the      │   EXIT
│ │ session      │─────────────────────▶
│ └──────────────┘ │
└──────────────────┘
```

Figure 6. Creation of new data files: the selection of the
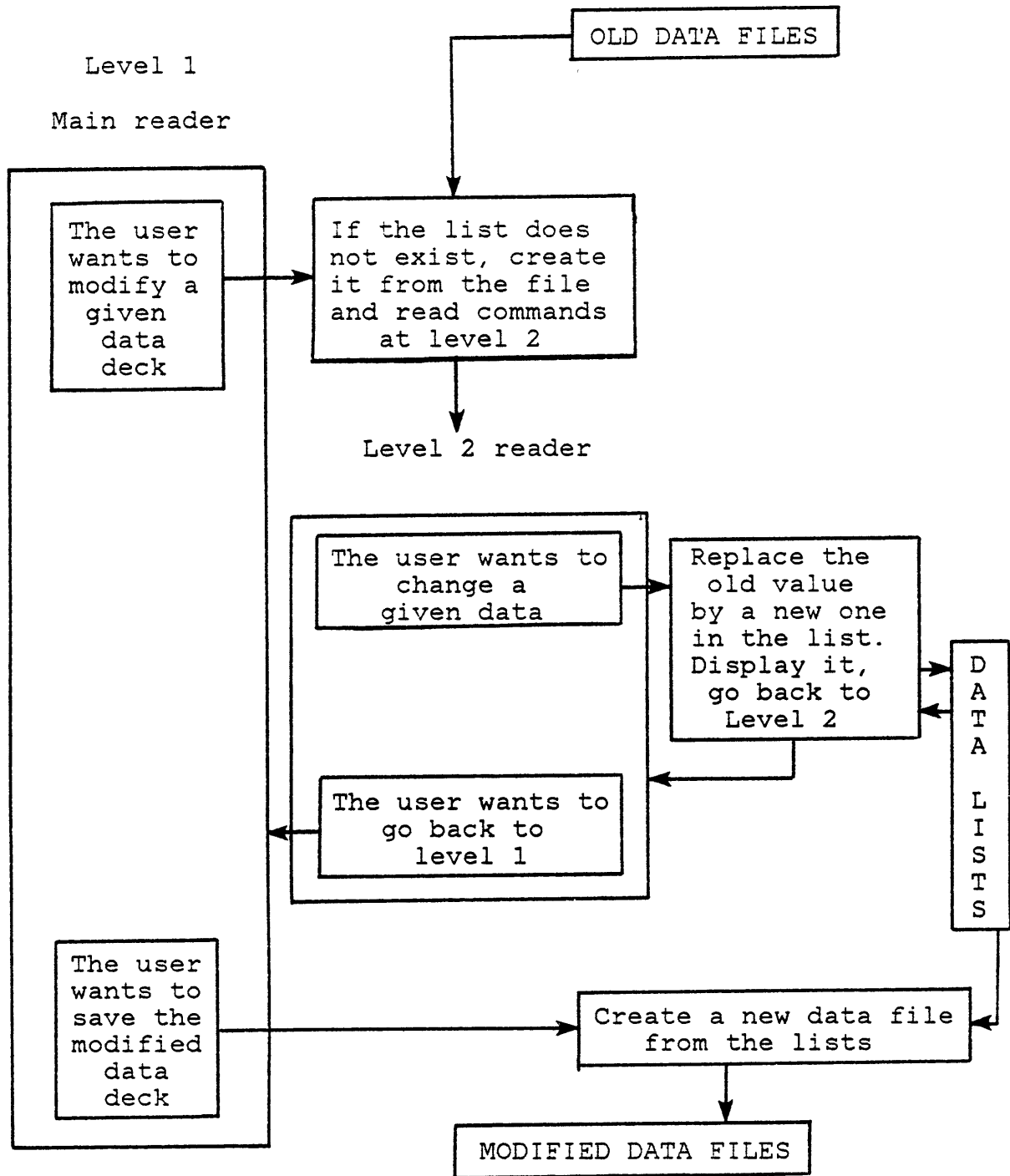   data deck is done at level one.

Figure 7. Modification of the old data: level 1 and level 2
are both described here. Level 3 is an option of
level 2.

With these two structures in mind it is now easier to understand the last figure (Figure 8 on page 51) which displays the whole organisation of the FA4 interactive program. All three levels of commands appear clearly, but by looking at the user's manual and the sample session in Appendices A and E, one can see that level three is different from the others for the following reasons:

- At level 1, the system is waiting for a command to be entered. To remind the user where he is, a prompt, 1.>, is printed.

- At level 2, we can be either in:

  a. the build command, which means the new prompts are specific to the data to be entered, or

  b. the modify command, where the computer is waiting for a command to be entered. Here, to remind the user, a prompt is printed, depending on which deck should be modified (e.g. ST.>, AC.>, etc.[21]

- Level 3 commands are more specific on what modification should be done on the data to be changed.

  For example we can change the data for city-pair AAA-BBB (level 2 command), and indicate at level 3 that the data to be modified is the distance. This procedure

---

[21] The reminding prompts make the system comply with rule 11 previously described (the user should never be lost).
ST stands for System data, AC for Aircraft data...

```
LEVEL 1

┌─────────────────────────────────────────────────────────────────┐
│ Main reader                                                      │
│                                                                  │
│        Commands                    Arguments                     │
│        _____                    _____                     │
│                                                                  │
│        BUILD                       System data      ST           │
│        MODIFY                      Airport data     AP           │
│        SAVE                        Aircraft data    AC           │
│        END                         City-pair data  CP            │
│                                    Route data       RT           │
└─────────────────────────────────────────────────────────────────┘
```
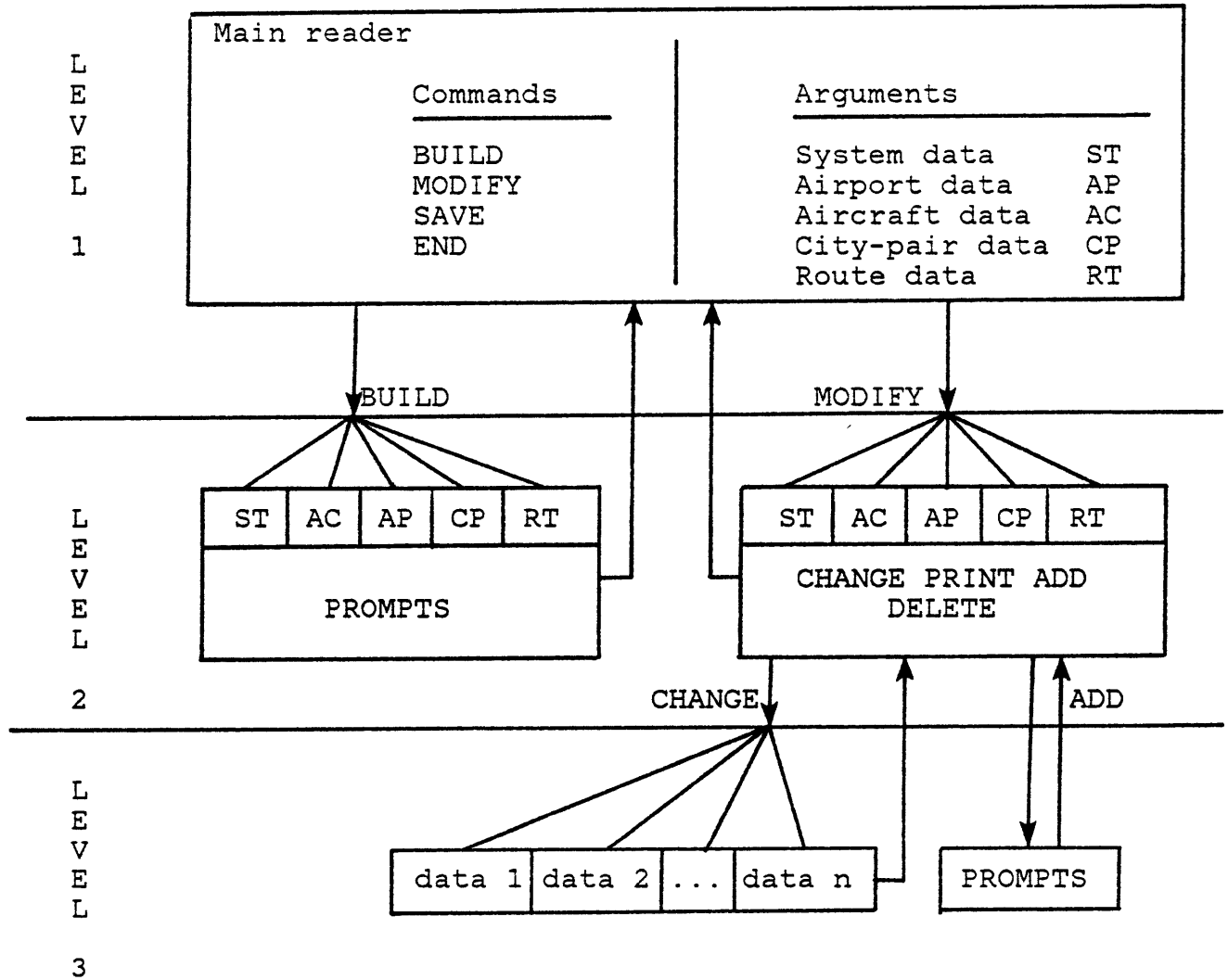
Figure 8.   Interactive   pre-preprocessor: only   level   3
            commands are not detailed.

is close to the system internal organization, since the computer will:

- first look for AAA-BBB in the list,
- then modify the corresponding distance.

However, in order to avoid complicated procedures, level 3 commands have been made options of level 2 commands, which means there are no specific prompts at level 3.

Finally, when the command ADD is used, the procedure is very similar to the creation of new decks since the prompts are the same, and corrections on the spot are permitted.

7.2 *SUMMARY*

The structure of the interactive pre-preprocessor has been designed to follow as much as possible the rules we discussed before. However, some elementary parts like the HELP command are missing. This is only because our system remains a *TEST SYSTEM* with which the real, final commands will be designed. In the mean time the global interactive FA4 structure is temporary since, again, it was designed as a prototype for further improvements and definitions.

The future system is discussed next.

8.0 THE FUTURE GLOBAL SYSTEM.

In this section the improvements and modifications which will be described correspond to a more efficient global system. They are not dictated, however, by experience and will certainly be subject to more improvements in the future.

8.1 THE PREPROCESSOR.

The separation of the pre-preprocessor from the preprocessor was done in order to keep the latter untouched. The current preprocessor is a working model that is familiar to the analyst. But a lot of operations executed in the preprocessor can and should be done directly by the interactive system.

1. In the actual interactive system data are read from disk files into working memory (lists). They are then rewritten into files and re-read again into working memory within the preprocessor. This is, of course, a total waste. The only way to avoid it is to rewrite the preprocessor in PL/1 and mix it with the pre-preprocessor so that it can access the lists directly.

2. Certain tasks which were executed at the preprocessor level should be done at the interactive input level. For example, each time a city-pair or a route was entered the

. preprocessor would check the city name code using the airport list. This should be done just when the user has entered a new city-pair or route, and in case of error should be rejected even before the list is created or modified.

3.   As we have seen before, the demand information of the city-pair deck can be entered two ways: either directly, or through the *CURVE GENERATOR*. In the interactive module the data required at the input are only the *direct* information: the *CURVE GENERATOR* has to be used externally. In the future system, two levels of prompting will be available:

* the prompt available at this time for direct input,

* a new prompt, asking for the data needed by the frequency-demand curve generator. That generator will also be rewritten in PL/1 and included into the new system.

Other commands can then be included such as -SKETCH-, to sketch a given frequency/demand curve (see part 3 for the curve definition). Options will be available to add capacity lines to the drawings (for different aircraft types and/or different maximum permissible load-factors) and eventually store these curves for printing.

This new stucture is summarized in Figure 9 on page 55 .

```
         ┌─────────────────────────────┐
    ┌───▶│  Data files created for     │◀─────────────────┐
    │    │      further use            │                  │
    │    │     (modifications...)      │                  │
    │    └─────────────────────────────┘                  │
    │                                                     │
    │  ┌──────────────────────────────────────────────┐  │
    │  │  Interactive system: command level            │  │
    │  │                                                │  │
    │  │  ┌──────────┐      ┌──────────────┐            │  │
    │  │  │  Curve   │─────▶│ Prompting and│            │  │
    │  │  │ generator│◀─────│   checking   │◀──Data entered
    │  │  └──────────┘      │     unit     │      from
    │  │                    └──────────────┘     keyboard
    │  │                           │                    │  │
    │  │                           ▼                    │  │
    │  │  ┌──────────────────────────────┐              │  │
    │  └──│   List generation unit       │──────────────┘  │
    │     └──────────────────────────────┘                 │
    │                    │                                  │
    │                    ▼                                  │
    │     ┌──────────────────────────────┐                 │
    │     │  Processing unit to          │                 │
    │     │      generate                │                 │
    │     │  LP. matrix format           │                 │
    │     └──────────────────────────────┘                 │
    └──────────────────  │
                         ▼
          ┌──────────────────────────────┐
          │  Output file formatted       │
          │       for the LP.            │
          └──────────────────────────────┘
```
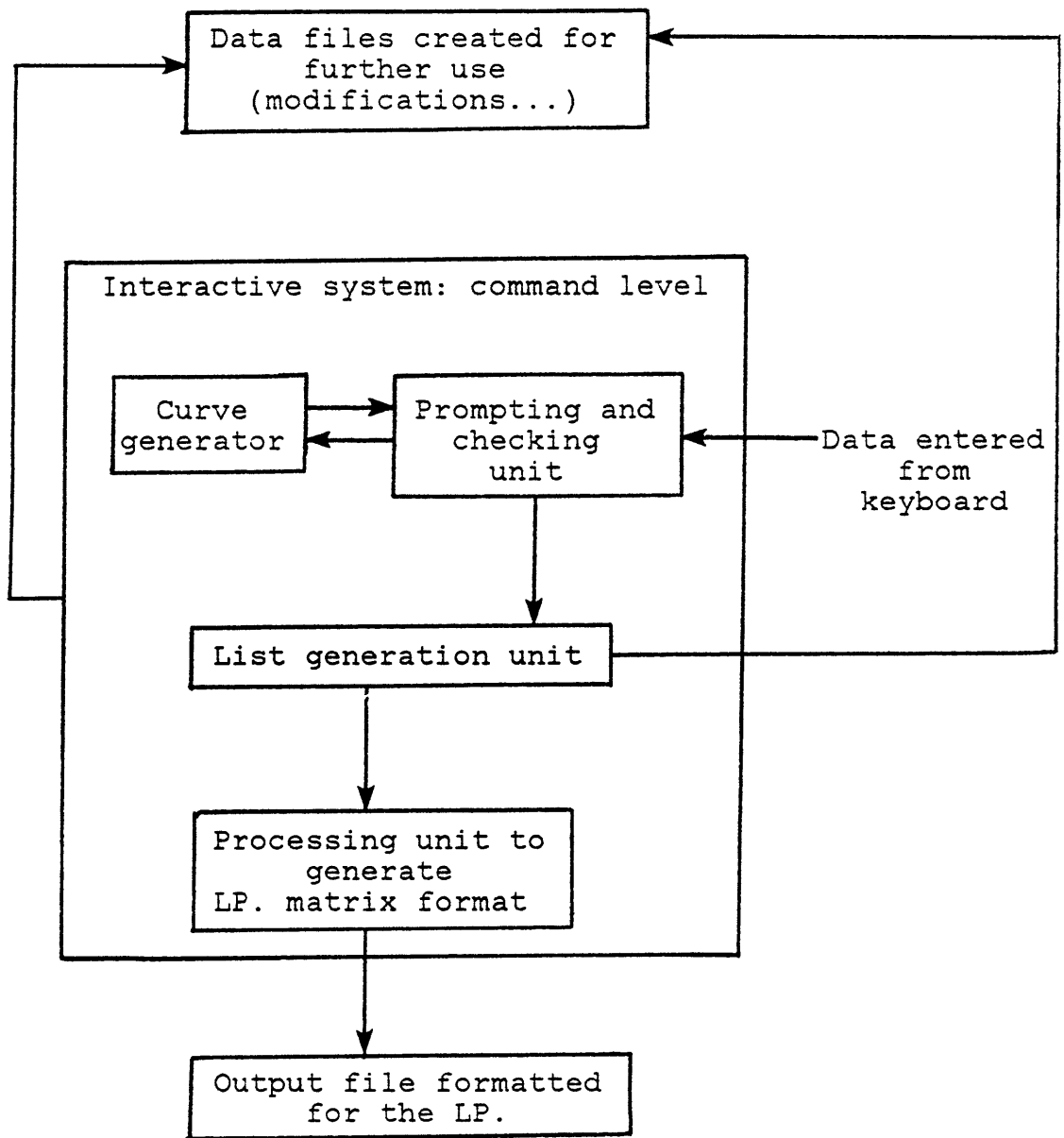
Figure 9. Overall structure of the future interactive FA4
           (1):   the preprocessor.

8.2 THE POSTPROCESSOR.

Here the task of modifying the postprocessor involves much more work. In the preprocessor case, the existing interactive system will be kept, even extended, and the old preprocessor included. Now everything has to be modified.

1. *The output format.* The existing output generated from the L.P. is suited only for direct printout. Now, we want to be able to read any section [22]of the results in any order, or perhaps get only a summary and finally a complete printed report.

    This flexibility means essentially one thing: we have to create a *list*. Each time a given report is needed, necessary data will be taken from the list, processed, and then formatted for a display on the terminal. That same formatted output might be stored on a file for further printing. The creation of the list allows any new computations on the results at the same time. Curves can be generated, for example.

2. *Compare the results of different runs.* This capacity was the other highly desirable option which was out of the question

---

[22]   These output sections correspond to those described in chapter 3.1 of this part (part two).

in the past (except by comparing the listings, which happened to be difficult with 2,000 city-pairs!.).

In order to get this option the previous "results-lists" have to be turned into disk files for storage. Then, when the user wants to compare his actual results with the old ones, these files are reaccessed and turned back into lists. In order to save time, only the interesting parts of the results will be processed. With the new lists ready, the user will be able to perform comparison tasks as well as statistics, trends, variations, etc. There should be at that step, *graph-commands* in order to make FA4 closer to a *GRAPHIC INTERACTIVE SYSTEM*. A graphic terminal, however, would be needed to fit completely into that category.

3. *Remain consistent and simple.* There will not be drastic changes in the commands syntax, which will remain very close to that of the preprocessor. Meanwhile, the organization of the output will not be different from the actual one. The division of the results will not change, only their display formats will be seriously modified in order to get something more condensed and easier to read when there are many city-pairs.

Details are given in Figure 10 on page 58.
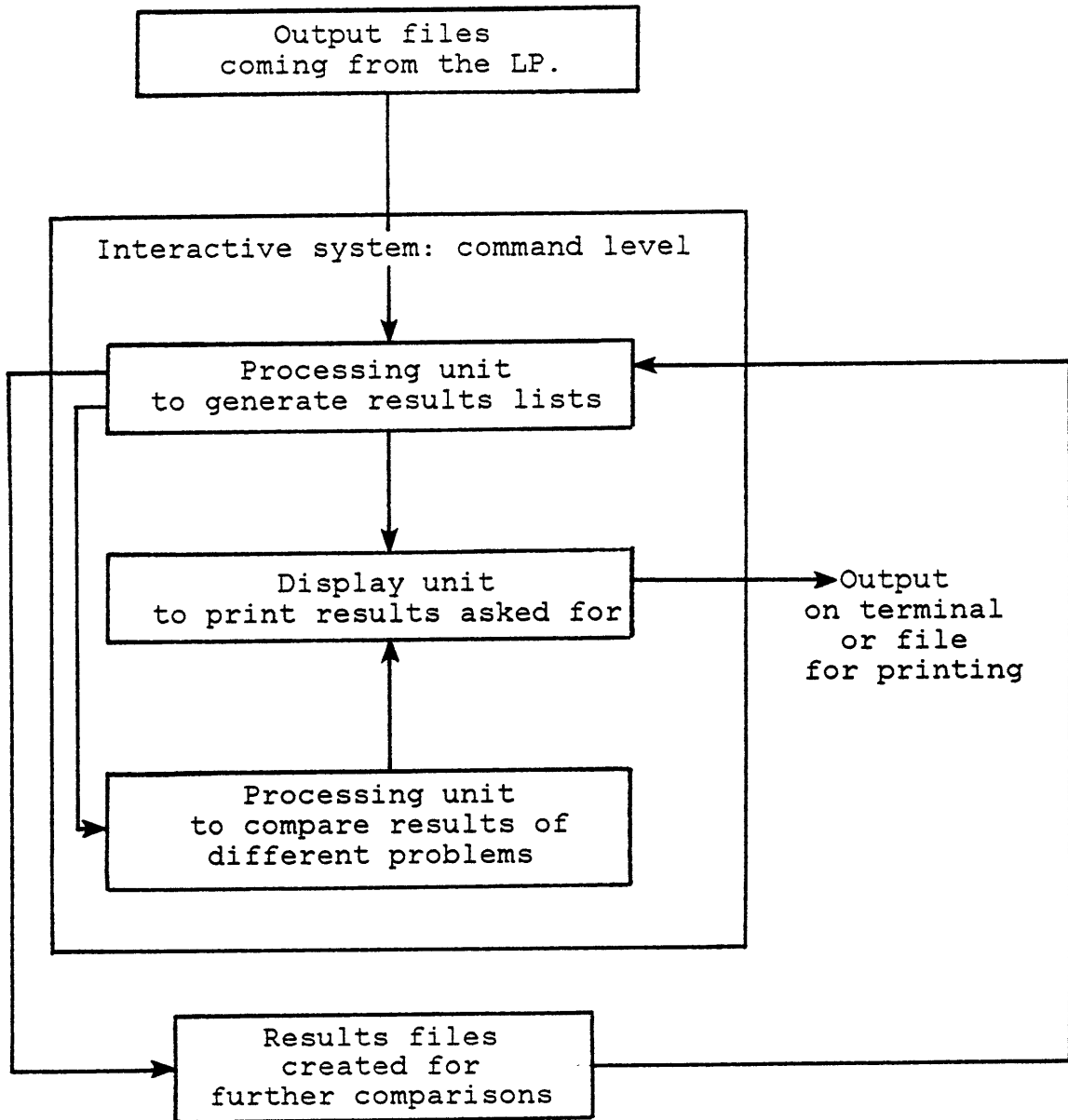
Figure 10. Overall structure of the future interactive FA4
            (2):   the postprocessor.

8.3 THE GLOBAL SYSTEM.

Of course, the pre- and post-processing interactive modules will not be separated (as they are today). There will be only one interactive environment. The only problem remaining then, will be the L.P. module.

For the time being, FA4 is implemented on IBM's CMS at MIT. The available L.P. module, called SESAME,[23] is interactive and accessible from CMS only. As long as this solver will be used, the operator will have to go back necessarily into the computer monitor system environment in order to solve the optimization. However, once inside the postprocessor module the user will not have to quit the PL/1 environment to come back to the pre-processing module. They will be in the same program. Figure 11 on page 60 is a summary of the overall organization of the future interactive FA4.

8.4 SUMMARY.

As with other interactive packages (see ref (23),(24)), a completely remade interactive FA4 can be used as a tool not

---

[23]   SESAME is an MIT designed program which replaces IBM's Mathematical Programming System Extended, MPSX.
SESAME stands for Systematized Extended Simplex Algorithm Module and Executor.
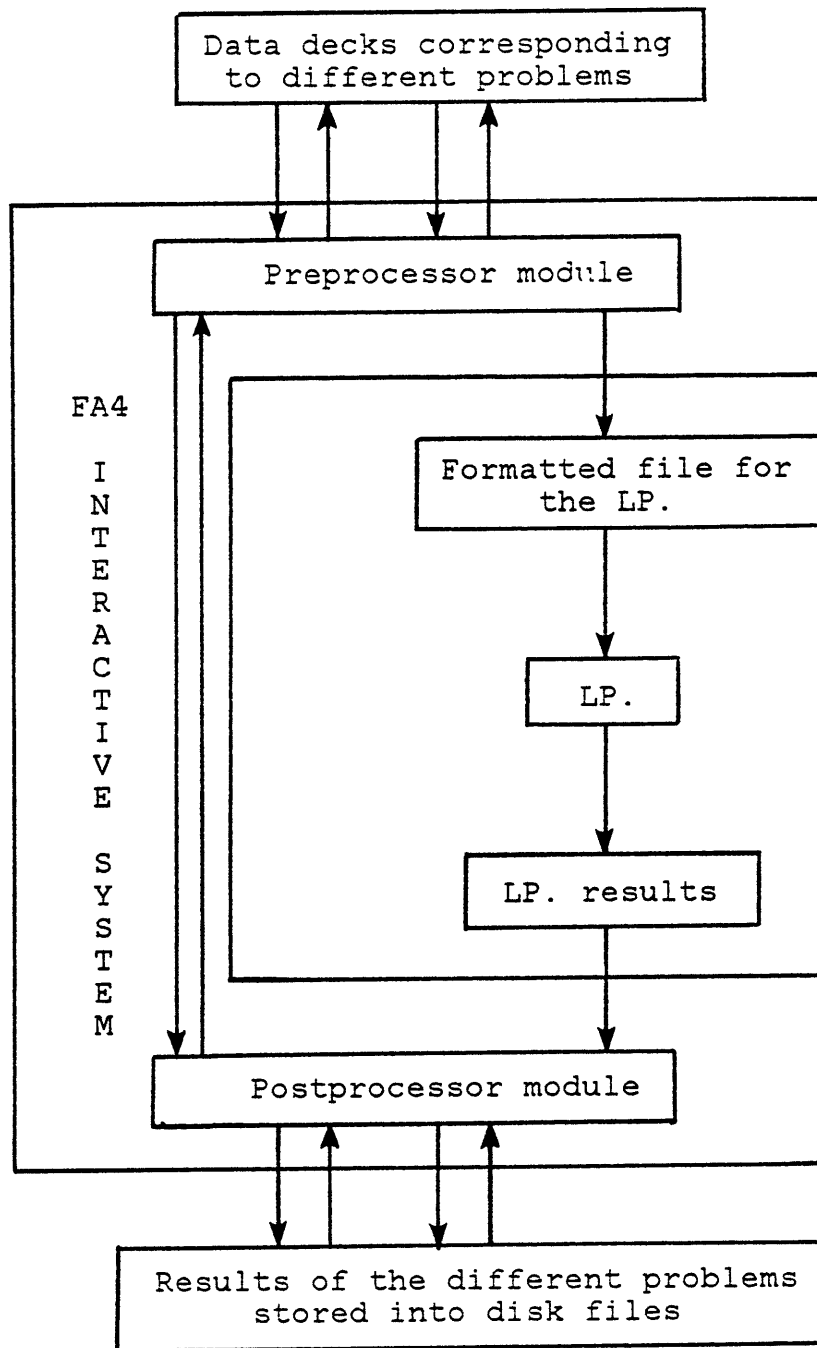
Figure 11. Overall structure of the interactive FA4
(3): the global system.

The future global system.

only by airline people but also by theoreticians in order to understand more completely the behavior of fleet assignments. Today, the linear formulation of the assignment problem may lead to incorrect results such as the phantom frequencies. This will be the subject of the next chapter.

9.0 *THE FREQUENCY-DEMAND CURVE*

This chapter is a general review of the demand, market share, and modal split models which underlie FA4. It covers all the results of research made on that subject within MIT's Flight Transportation Laboratory.

The following section will explain how is generated the equation of the demand for air transportation in general (all carriers included) on a given market. The demand provided to FA4, however, should correspond to an individual airline. Therefore, the next section will show how individual demand is derived from the general equation.

9.1 *THE GENERAL MATHEMATICAL MODEL*

The general equation of the demand for transportation between cities p and q using airlines is:

$$D_{pqa} = K \left(\frac{P_p \, P_q}{d^2_{pq}} \right) \left(\frac{C^\alpha_{pqa} \, T^\beta_{pqa}}{\sum_m C^\alpha_{pqm} \, T^\beta_{pqm}} \right) \qquad (9)$$

$$\underset{\text{gravity model}}{\blacktriangle} \qquad \underset{\text{modal split model}}{\blacktriangle}$$

The gravity model (see ref (2)) relates demand for market PQ to the respective populations of the cities P and Q (P_p and P_q) and the square of the intercity distance d^2_{pq} through a constant K. The formulation is the same as *Newton Gravity Law*.

The frequency-demand curve   63

When a fleet assignment study is done, the time period considered is usually small. It means that the populations of both cities can be considered fixed. The gravity model will then be expressed in our formula as a constant. Demand for airline service becomes:

$$D_{pqa} = K_g \left(\frac{C^{\alpha}_{pqa} \, T^{\beta}_{pqa}}{\sum_m C^{\alpha}_{pqm} \, T^{\beta}_{pqm}} \right) \qquad (10)$$

The modal split model relates here demand for air transportation to the cost of travel between P and Q using aircraft as a general mean (C_{pqa}) and the time it takes (T_{pqa}), to the costs and times using other means (C_{pqm} and T_{pqm}). In this formula α and β are negative elasticities which determine the influence of time versus costs. Notice that α and β can be made mode dependent which gives latitude to rank the influence of all types of transportation. In that case the formula would be:

$$D_{pqa} = K_g \left(\frac{C^{\alpha_a}_{pqa} \, T^{\beta_a}_{pqa}}{\sum_m C^{\alpha_m}_{pqm} \, T^{\beta_m}_{pqm}} \right) \qquad (11)$$

Dealing with air transportation, we may be considering only a small part of traffic between cities P and Q. Thus, if we change air fares or trip time, the total sum Σ for all modes of trans-

The frequency-demand curve   64

portation, will not change very much. It is almost a constant and allows us to simplify the demand equation.

Finally, it is possible to consider travel for which fares are fixed. C_{pqa} is then a constant also.

At this point the initial demand equation for airline traffic becomes simply a function of travel time:

$$D_{pqa} = K_{pqa} \; T_{pqa}^{\beta_a} \tag{12}$$

Before going any further, it must be underlined that the modal split model is really used here only for its mathematical convenience. As W. SWAN says[24]:

"By including enough dummy constants ... the modal split model can achieve an arbitrarily accurate curve fit to whatever general market behavior actually exists ... It would be foolish to go any further then stating that the general behavior pattern makes sense."

Moreover, the modal split model used here cannot show interpenetration of markets (an improved air service affects more trains than buses) and market share of two very close modes (to include separately propeller and jet aircraft would overestimate the total air market share). It is only good at splitting a market betwen two completely different modes: aircraft and autos for instance.

---

[24]   see reference (35)

Let us see now how travel time using air transportation T_{pqa} is expressed. The general assumption is:

$$T_{pqa} = T_B + T_D \qquad (13)$$

where

- T_B is the flight block time (T_{BL}) + the time spent in the airport + the average surface travel time to and from that airport.

- T_D is the displacement time, that is by how much a passenger is displaced forwards or backwards from his ideal departure time because of the existing global schedule (all airlines included).

In a 1977 study Steven ERIKSEN from MIT's Flight Transportation Laboratory gave an algorithm to compute the values of T_D, knowing the passengers ideal desires and departure times available.[25] In our case, however, since the fleet assignment does not include time of day, another expression was proposed:

$$T_D = K \, \frac{T_A}{F} \qquad (14)$$

where T_A is the time length of one day of activity (from the first to the last departure) and F the total number of direct

_____

[25] For more details see ref (5)

flights available on the market (non-stop frequency for all airlines). K is a constant.

The general rule was K = 0.25 and T_A = 16 hours, leading to:

$$T_D = \frac{4}{F} \qquad (15)$$

In 1977 Professor MATHAISEL compared the T_D's computed with this formula to those produced by Eriksen's algorithm [26] He showed that, in fact, a better value was :

$$T_D = \frac{5.68}{F} \qquad (16)$$

In the frequency demand generator currently available in FTL, the formula used is T_D = DAY/F (DAY being fixed by the user). At that step the model equation becomes:

$$D_{pqa} = K_{pqa} \left(T_B + \frac{DAY}{F} \right)^{\beta_a} \qquad (17)$$

where DAY is selected by the user.

What about F? Two problems must be taken care of:

1. *Overlapping* . In a competitive environment, two airlines serving the same market, might schedule two flights at the same time. Even if the real frequency is 2, it looks as if there were only one service available (time wise) for the

---

[26] See ref (21)

passengers. The total frequency has then, to be decreased
to take into account the effect of these overlappings
throughout the day on the total market demand fonction.

Many solutions were suggested, we use the one proposed by
W.Swan which is, as Professor MATHAISEL also shows in his
report, the best suited to the U.S domestic market:

$$F = \frac{\text{Total real frequency}}{1 / \sqrt{\max_{c} (MS_c)}} \qquad (18)$$

$\max (MS_c)$ is the biggest market share among the competitors
(c) on the market.

This formulation is the first way to take into account com-
petition in the frequency-demand model. But overlapping is
a factor which affects all airlines in the same way. For
this reason, it is included in the general equation.

2. *Multi-stop flights* . It is obvious that the degree of attrac-
tiveness of a flight decreases when the number of interme-
diate stops increases. Since our demand equation requires
only one frequency, we consider that non-stop services are
kept untouched but multi-stop flights are reduced to an
equivalent number of non-stop flights by the use of a fre-
quency weighting factor.(If one-stop flights are 50% less
attractive, we need two of them to make one eqiuvalent
non-stop flight.) Our total real frequency becomes then:

$$N^m = K_0^m \, r_0^m + K_1^m \, n_1^m + \ldots \qquad (19)$$

k_s^m is the frequency weighting factor corresponding

to s-stop flights on market m (pq here) where:

$0 < k_s^m < 1$ for all $s > 1$

n_s^m is the number of s-stop flights

Let us see what our general airline demand equation is, at this

stage:

$$D_{pqa} = K_{pqa} \left(T_B + \cfrac{DAY}{\cfrac{K_0^m \, n_0^m + K_1^m \, n_1^m + K_2^m \, n_2^m + \ldots}{1 \, / \, \sqrt{\max_c \, (MS_c)}}} \right)^{\beta_a} \qquad (20)$$

This is the final general form, from which the frequency-demand

curves are computed for individual airlines in the FA4 (and

FA7) curve generator.

9.2 THE INDIVIDUAL TRAFFIC-FREQUENCY CURVE GENERATION

Usually, when we want to generate a traffic-frequency curve

for a given airline and a certain market, the information used

(and available) is at a given date:

- the total level of service for all carriers n_0^m, n_1^m, n_2^m ...

- the total demand D_{pqa}

- the block times T_{BL} ...

- the market shares for all carriers and especially: the market share of the biggest carrier on the market and our airline market share.

The shape of the traffic-frequency curve is already fixed by the formulation used for the general demand equation. The above values will calibrate the "amplitude" of that curve.

1. Computation of the Frequency Weighting Factors

   In the current traffic-frequency curve generator, the k_s^m's were derived with the following assumptions:[27]

   - a flight attracts people within one hour around its departure time.

   - the active day is 16 hours.

   Any k_s^m is then established as the ratio of the remaining number of hours of the active day during which an s-stop flight can be scheduled without being in an attraction period of an r-stop flight (r<s), to the number of hours of the active day during which flights can be scheduled. (16 hours $-T_{BL}$) Where T_{BL} is the block time of the flight.

   _____

   [27]   See ref (31)

For example:

If we have in a market m, n_0^m non-stop flights, taking into account overlapping, the effective non-stop frequency will be:

$$\frac{n_0^m}{1 \ / \ \sqrt{\max_c \ (MS_c)}} \qquad (21)$$

each of these non-stop flights will then attract passengers on

$$\frac{n_0^m}{1 \ / \ \sqrt{\max_c \ (MS_c)}} \quad \text{hours of the day, leaving}$$

$$16 - T_{BL} - \frac{n_0^m}{1 \ / \ \sqrt{\max_c \ (MS_c)}} - 1 \quad \text{hours for the}$$

one-stop flights to be usefully scheduled (-1, was added after calibration, showing the undesirability of having a one-stop during the last hour of the day useful for non-stops). Therefore k_1^m will be :

$$k_1^m = \frac{16 - T_{BL} - (n_0^m \ / \ (1/\sqrt{\max_c \ (MS_c)})) - 1}{16 - T_{BL}} \qquad (22)$$

since n_1^m is known historically, k_2^m can be computed with the same procedure ...

The frequency-demand curve

As soon as k_s^m is < 0, it is set to 0 as well as all the k_t^m for t > s.

2. Computation of the market constant K_{pqa}

Since:

- D_{pqa} for our market is given
- max (MC_c) is known as well as T_B
- DAY can be set at 5.68 as proposed by Professor MATHAISEL.
- β_a is set usually close to -0.5

it is possible by reversing the general formula to compute the market constant:

$$K_{pqa} = \frac{D_{pqa}}{\left(T_B + \dfrac{DAY}{\dfrac{K_0^m n_0^m + K_1^m n_1^m + K_2^m n_2^m + \ldots}{1 / \sqrt{\max_c (MS_c)}}} \right)^{\beta_a}} \tag{23}$$

3. Computation of the demand for an individual carrier, c

We are at the final stage. We want the value of the demand for our airline (c) at various frequencies: $D_{pqc}(n_c^m)$, n_c^m being the equivalent frequency of flights by carrier c. Here another assumption is made:

Frequency share = Market share

The frequency-demand curve

The usual frequency share-market share curve which is often described by a flat S shape, is assumed to be a straight line:

% of total market share MS_c
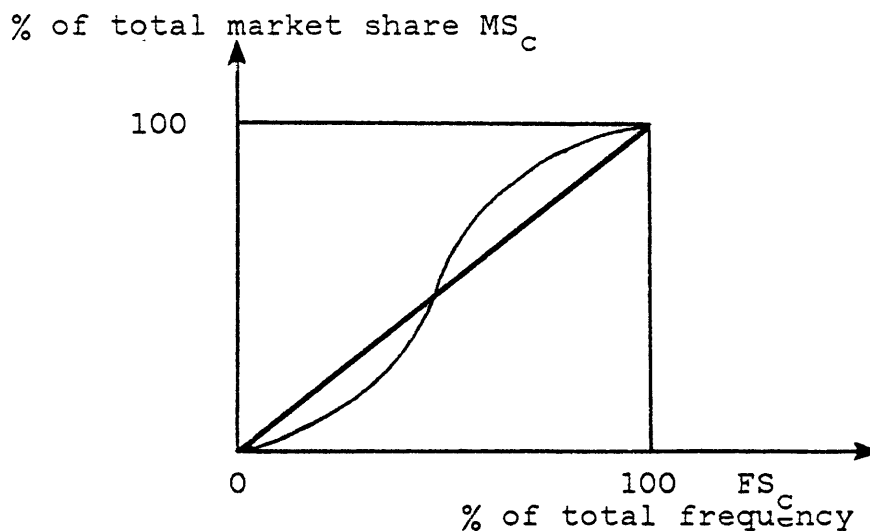
Since we know the market share of airline c, MS_c, we know the market share of its competitors, $(1 - MS_c)$, which is their frequency share with our assumption. The total equivalent frequency for the competitors will be, then:

$$F^m_{comp} = (1 - MS_c) \; \frac{K^m_0 \, n^m_0 + K^m_1 \, n^m_1 + \ldots}{1 \; / \; \sqrt{\max_c \, (MS_c)}} \qquad (24)$$

Assuming that the competitors frequency remains unchanged, it is possible to calculate the new total equivalent frequency on the market when carrier c offers n^m_c services:

$$F = F^m_{comp} + n^m_c$$

Individual demand for airline c can now be expressed:

$$D_{pqa}(n_c^m) = \left(\frac{n_c^m}{\dfrac{K_0^m\, n_0^m + K_1^m\, n_1^m + \ldots}{1 / \sqrt{\max_c (MS_c)}}} \right) K_{pqa} \left(T_B + \frac{DAY}{F_{comp}^m + n_m^m} \right)^{\beta_a} \quad (25)$$

$\blacktriangle$ our new market share = our new frequency share.

$\blacktriangle$ The new total airline demand on the market.

Finally it must be pointed out that the equation discussed above has always the following shape:
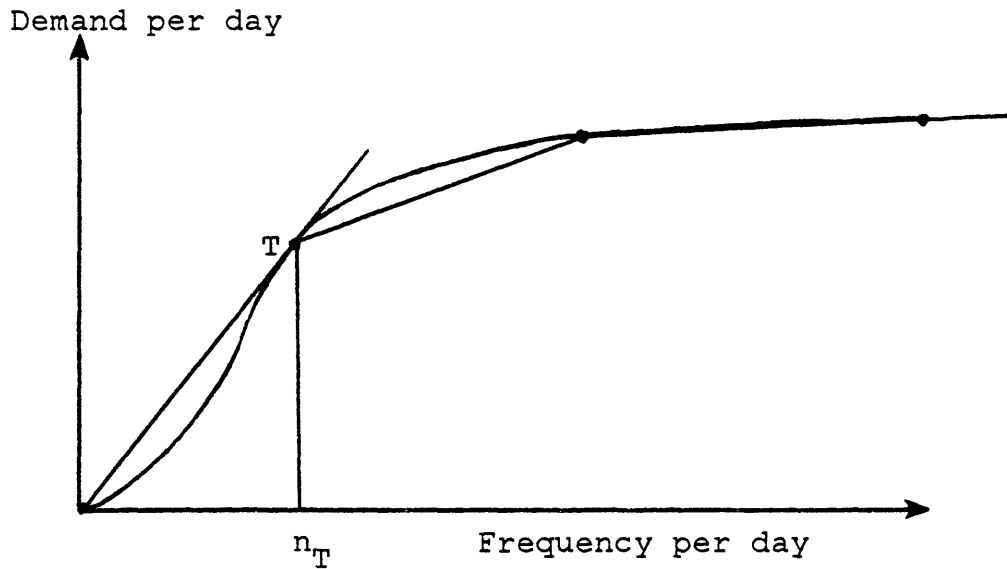
Since this curve must be entered into the L.P. formulation in a piecewise linear fashion, with a continuous convexity,[28] the first "break-point" should be after T, the point where the ray coming from O is tangent to the curve. This restriction on the

---

[28] This continuous convexity will ensure that the segments of the curve are "filled" by frequencies in proper order (e.g. frequency on segment i remains null as long as frequency on segment (i - 1) has not reached its upper limit).

first break-point is not a problem for normal or high levels of demand where the frequency corresponding to T, n_T is less than normal frequencies of service. On the other end, at low levels of demand n_T might be higher than normal service frequency and the user should be careful (The current code for FA4 checks anyway for negative curvature).
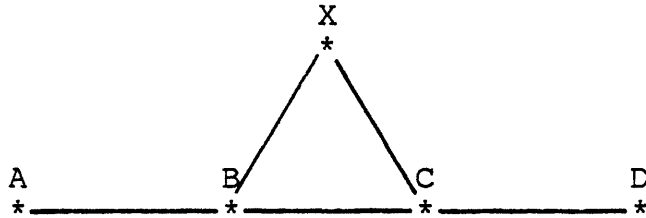
9.3 *SUMMARY*

This revue points out two essential things:

- in the business market, people are sensitive to their travel time, which they want to minimize.
- non-stop flights are chosen before one-stops (which are themselves chosen before two-stops etc...) as shown in the frequency weighting factors computation. This second fact will be used in the chapter 11, in order to improve the answer to the phantom frequency problem.

10.0 *THE PHANTOM FREQUENCY PROBLEM*

Rather than using immediatly the complete notations defined in Appendix B and used in Part 1, let us define a small case problem with simplified symbols.

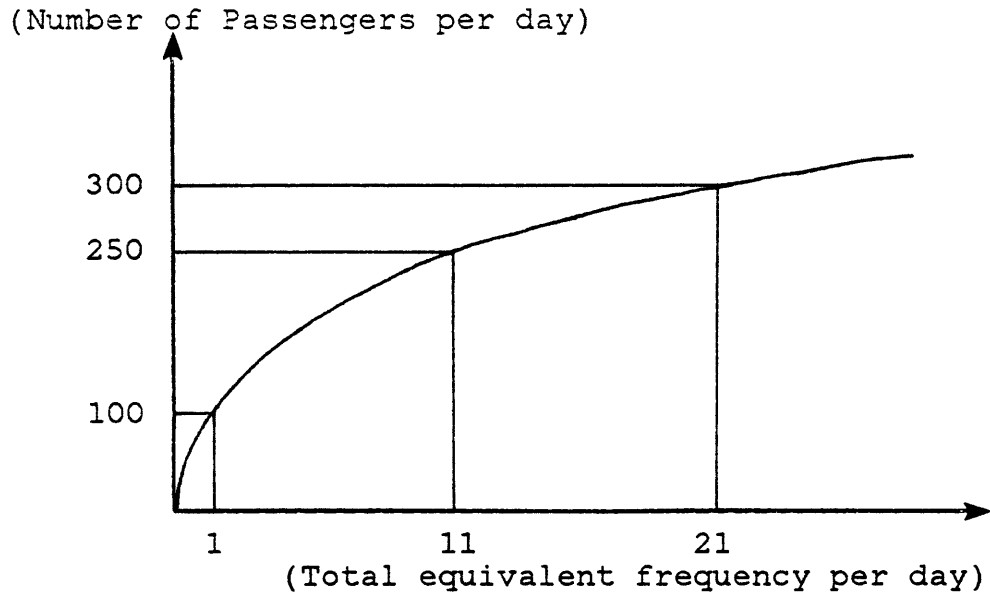At first, we have a small network with five stations A,B,C,D and X.

```
                         X
                         *
                        / \
                       /   \
                      /     \
          A        B /       \ C        D
          *_____*_____*_____*
```

The only possible routes we are allowed to fly (roundtrip) are:

1. BC

2. BXC

Let us also suppose that our demand is as follows:

1. There is, for the time being, no demand from A to B, C to D, A to X and X to D.

2. There is a substantial demand from B to X,

3. and substantial demand from X to C.

4. The demand between B and C corresponds to the following curve:

(Number of Passengers per day)

```
           ↑
           │
       300 ├─────────────────────────────────────────●
           │                                      ╱
       250 ├──────────────────────────●        ╱
           │                       ╱
           │                    ╱
           │                 ╱
           │              ╱
       100 ├──────●    ╱
           │      │ ╱
           │     ╱│
           └─────┴────────────┴───────────────┴──────────→
                 1            11              21
                 (Total equivalent frequency per day)
```

And finally the frequency weighting factors on BC are:

•   non-stop frequencies K = 1.0

•   one-stop frequencies K = 0.5

The result of this problem will be:

•   passengers flying from B to X will be on the first segment of route 2.

•   passengers flying from X to C will be on the second segment of route 2.

- passengers flying from B to C will fly routes 1 (non-stop service) or 2 (one-stop service).

Let us now imagine the following extreme solution which satis-fies economic optimization.

Route	Frequency	Aircraft capacity	Average load-factor
BC	1 flight	250 seats	100%
BXC	20 flights	100 seats	100%

And:

- Segment BX of route 2 is filled only with the demand on mar-ket BX (which is substantial).
- Segment XC of route 2 is filled only with the demand on mar-ket XC (which is also substantial).
- The result for route 1 (BC) is obtained as follows:

  The total equivalent frequency on market BC is:

  $$(1.0 \times 1) + (0.5 \times 20) = 11$$

  flight 2   flights 4

  The demand curve, described before leads to 250 passengers which are all carried by route BC.

It means that route BC carries a demand which was generated by route BXC. Since the 10 equivalent frequencies of BXC cannot carry BC passengers there should not be counted to estimate the

demand on route BC. The 10 extra frequencies are called "phan-tom frequencies".

One might complain, here, that in real life passengers have access to scheduled flights on a first-come first-served basis. Therefore, there is no reason for BC passengers not to be accepted on BXC flights. In fact, FA4 is an optimization problem and since it is more profitable for the system to carry separately BX and XC passengers, rather then BC passengers on route BXC (the fare for BC passengers does not include the cost of landing in X), BC passengers will be rejected from BXC if the demand on markets BX and XC is enough.

In the previous example the problem arose because:

1. It is profitable to serve market BC.
2. Market BC is served by a non-stop route (BC) and a multi-stop route (BXC).
3. Even if routes BX and XC were included, the system would choose route BXC to carry markets BX and XC demand:
    *   it does not cost more since both segments BX and XC can serve that demand non-stop.
    *   it "creates" at no cost new one-stop frequencies (for market BC) which will inflate the demand carried on direct route BC and, then, increase the yield.

But there are other examples. If the routes available had been:

The Phantom Frequency problem

1. BC

2. ABC

3. BCD

we can imagine another phantom frequency case:

* The demand on AC and BD is substantial.

* The demand on BC is the same as before.

* No demand on AB and CD.

The solution could have been:

Route	Frequency	Aircraft capacity	Average load factor
BC	1 flight	300 seats	100%
ABC	10 flights	100 seats	100%
BCD	10 flights	100 seats	100%

Here segment BC of routes ABC and BCD is filled with AC and BD passengers. BC passengers are only carried on route BC (frequency=1). But as we can guess, the total equivalent frequency computed was 21 instead of 1:

$$1 + 20 = 21$$

non-stop flight   non-stop flights
        with   with
    route BC   routes ABC & BCD

leading to 20 phantom frequencies and 200 extra passengers.

The reasons for these extra frequencies are the following:

- It is profitable to serve market BC.

- Market BC is served non-stop by more than one route.

- To a lesser extent, there is a mismatch of routes and demand: non-stop routes AC and BD should be included. However, even if these routes were added the system might decide to fly ABC or BCD anyway in order to carry AC or BD passengers, because the yield generated by the increase in market BC demand (due to more BC frequencies) can offset the cost of flying AC or BD passengers with one stop.

   The incentive to fly ABC or BCD will even be higher if there is a demand on markets AB or CD.

In reality, it is very difficult to figure out to what degree the phantom frequencies are present in a larger problem. There might be, for example, space remaining on route BXC for BC passengers. This will lessen the gap between computed and real demand, but will still inflate the total number of passengers carried.

With these two examples, we can draw already a few conclusions:

1. The problems can occur when a given market is served by more than one route.

2. There are two possibilities for a market to be served by more than one route:

- The market appears everywhere as a non-stop:

    BC ABC BCD etc... (second example)

- The market is also served by multi-stop flights:

    BXC BXYC etc... (first example)

3. There could not be phantom frequencies in FA3 where multi-stops routes did not exist. (None appeared in FA1 and FA2 as well, but only because demand was not frequency dependent.)

10.1 *THE PREVIOUS ANSWER*

FA6, as we have seen, was the first model to include an answer to our problem.[29]

With simplified notations, the demand generated in FA4 on each market was :

$$D^m = f(N^m) \tag{26}$$

f is the frequency demand curve and N^m the total equivalent frequency:

---

[29] See ref (10)

$$N^m = K_0^m \, n_0^m + K_1^m \, n_1^m + \ldots$$

$$= \sum_s K_s^m \, n_s^m \qquad\qquad (27)$$

n_0^m is the number of non-stop flights

n_1^m is the number of one-stop flights

etc...

Demand in pax/day

$$N^m = \sum_s K_s^m \, n_s^m \quad \text{Equivalent frequency/day}$$
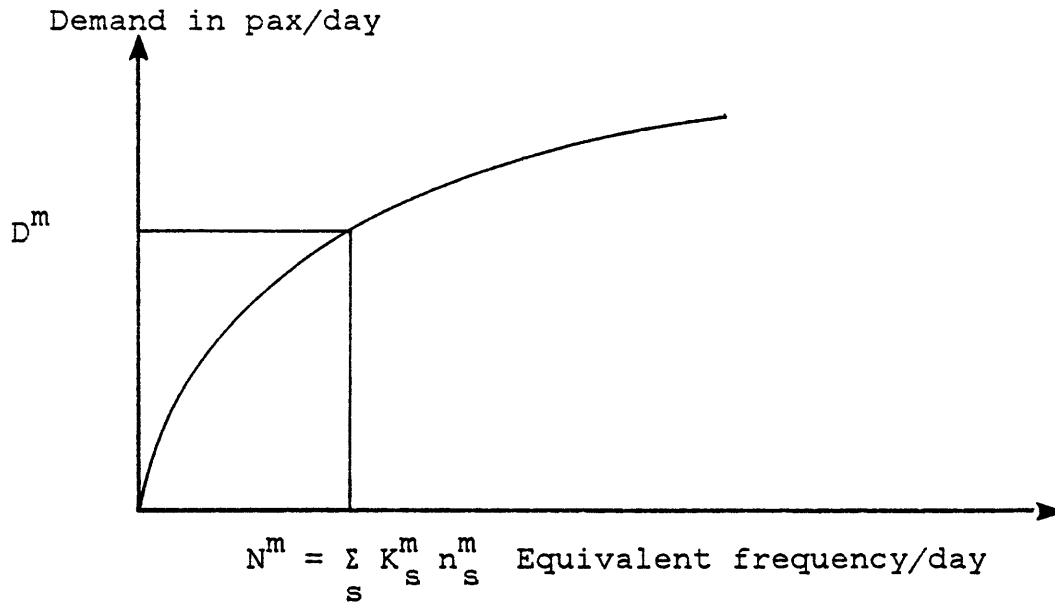
The *total* number of passengers carried on *all* routes serving market m should, then, be less or equal to that *total* demand D^m.

So, in FA6 it was decided that the demand would be split by routes serving the market:

$$D_{r^m}^m = f(n_{r^m}) \qquad\qquad (28)$$

r^m being a route serving market m

and the traffic carried will be limited *individually on each route* by this new expression of the demand.[30] Other constraints would also be added.

Suppose we have two flights serving the same market non-stop (market AB served non-stop by flights AB and ABC for example) and the following curve:

D^{AB} Demand (pax/day)
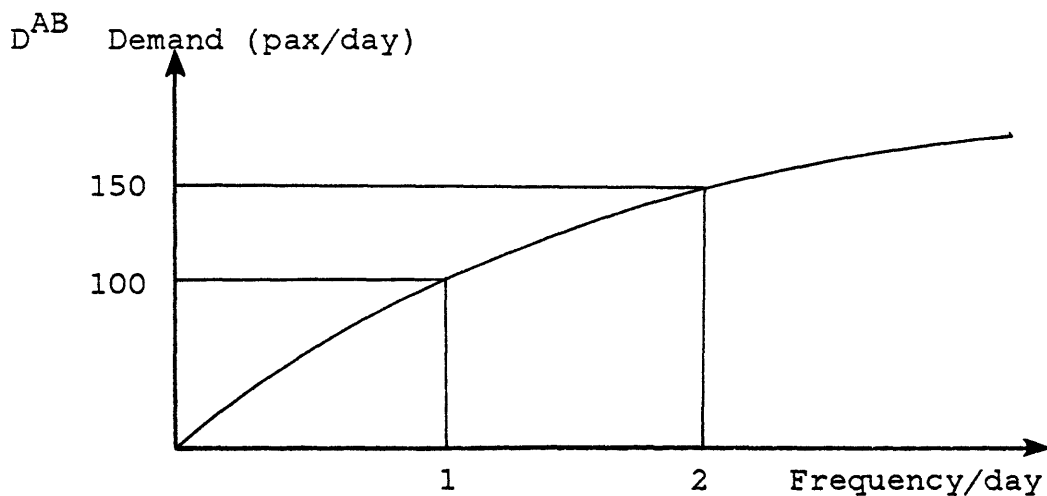
the demand in market AB would be:

$$D^{AB}_{AB} = f(1) = 100 \quad \text{this restrains the demand in market AB}$$
$$D^{AB}_{ABC} = f(1) = 100 \quad \text{from routes AB or ABC from being inflated}$$

the total demand is 200, which is wrong since the curve is convex. From the curve shown, when there is a total of 2 flights,

---

[30] The following discussion explains how the demand was handled in FA6. The fleet assignment model equations, however, include the traffic effectively carried, which is less or equal to this demand.
   In fact, each time a new type of *demand* is introduced, the corresponding *traffic* equation is created.

market AB attracts only 150 passengers which means we should keep also the old constraint:

$$D^{AB} = f(n_{AB} + n_{ABC})$$

If there were three routes (AB, ABC, ABD) serving the market with one flight each, we would have:
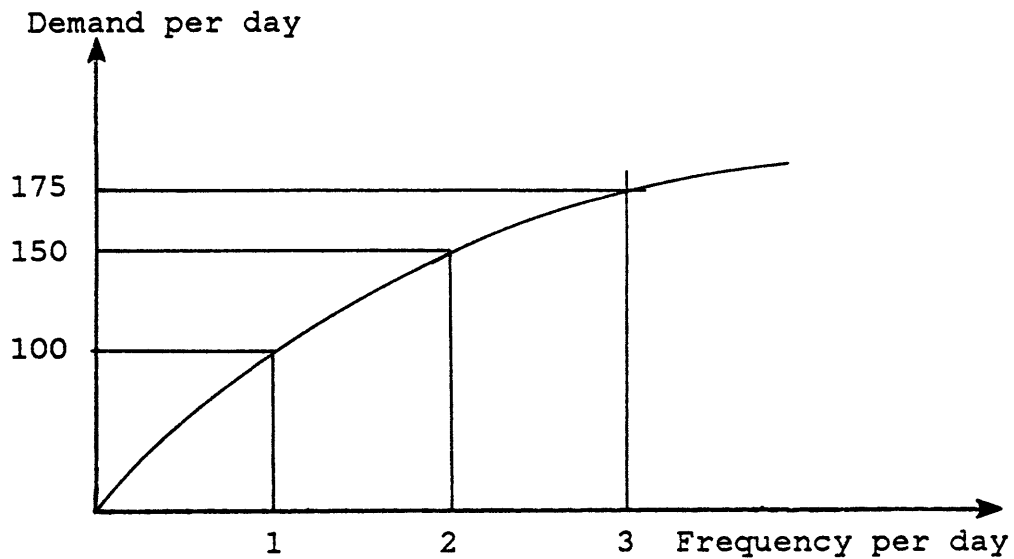
Demand per day

total demand: $D^{AB} = f(n_{AB} + n_{ABC} + n_{ABD})$
$$= f(3) = 175 \text{ pax per day}$$

individual market demand by route:

$$D^{AB}_{AB} = f(n_{AB}) = f(1)$$
$$D^{AB}_{ABC} = f(n_{ABC}) = f(1)$$
$$D^{AB}_{ABD} = f(n_{ABD}) = f(1)$$

But now, the phantom frequency problem can occur between groups:

If ABD was saturated with AD passengers, the demand it gener-
ated can be carried on (AB + ABC). ABD frequency becomes a phan-
tom frequency for market AB and inflates the total demand
carried on routes AB and ABC.

We have then to put a limit on market AB demand carried on
these two routes:

$$D_{AB+ABC}^{AB} = f(n_{AB} + n_{ABC})$$

The same thing of course applies for route combinations (AB +
ABD) and (ABC + ABD).

Finally, if there are n routes serving market m, we have to
write the demand-frequency relationship, route by route, by
groups of two, three,..., n. This is practically impossible
since the number of equations needed grows very quickly.

The suggestion in ref (10) was to solve the optimization,
check the results to see if there were "cases of inflated
demand", add the equations described only to the corresponding
markets, and re-run the optimization.

However, this is not very practical. The new solution sug-
gested, as we will see, decreases the number of equations
needed.

11.0 *ANOTHER SOLUTION TO THE PHANTOM FREQUENCY PROBLEM*

11.1 *THE NEW ASSUMPTION*

As we have seen in chapter 10 on phantom frequencies, the phe-
nomenon could occur each time a given market was served by more
than one route. No distinction was made at that time between
routes serving that market non-stop, and routes serving it with
one-stop, two-stops etc... After the review made, we will now
assume that:

Business passengers will generally start to choose non-stop
flights and then one-stop, two-stops etc... Let us see with an
exemple what it means.

Suppose we have a market m with a total equivalent frequency:

$$N^m = K_0^m \, n_0^m + K_1^m \, n_1^m + \ldots$$

n_0^m and n_1^m being the number of non-stop and one-stop flights on
market m (overlapping factor included).

The frequency-demand curve will be:
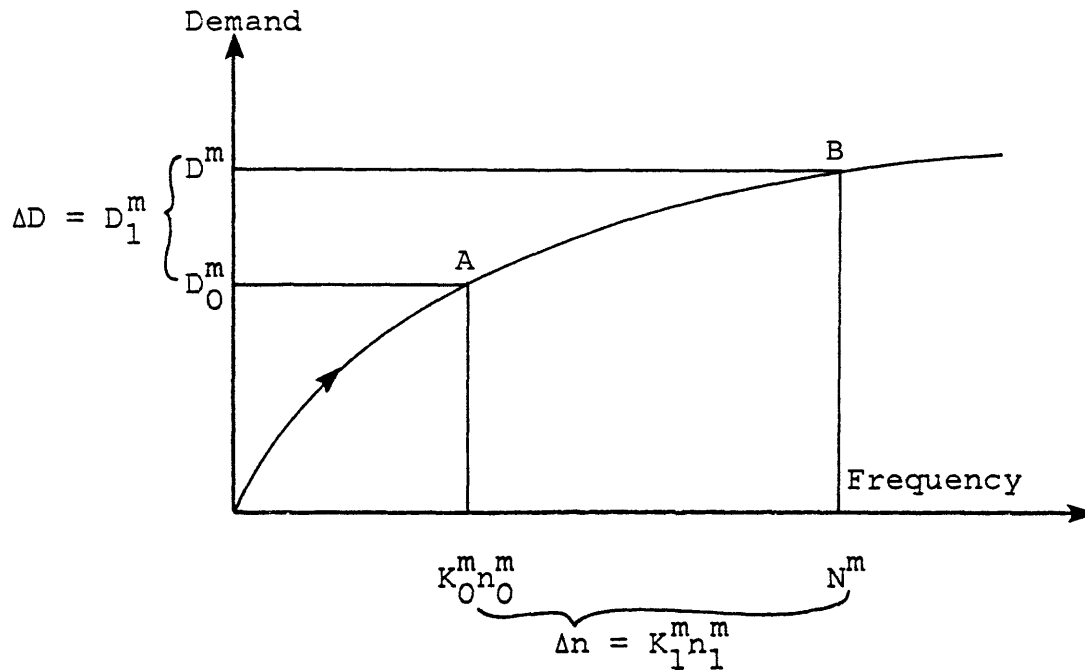
The total demand generated is D^m, and the previous assumption means that since demand will start on non-stop flights, we can follow the curve in the direction of the arrow.

- when we reach point A, we have the demand for non-stop flights D_0^m corresponding to n_0^m.

- then, we start looking at the demand for one-stop flights which accumulates up to D^m when we reach N^m.

  - the one-stop frequency gap covered was $\Delta n = K_1^m n_1^m$

  - the one-stop gap covered was $\Delta D = D_1^m$ (demand for one-stop flights) which appears after demand for non-stop flights.

11.2 *THE FORMULATION*

Let us first review certain notations which will be used (for the others see Appendix B).

- n_{mi} are the demand curve break point frequencies
- n^{mi} represents the frequency we have reached on segment i of the curve (if n^{mi} is not null, then all n^{mj} for $j<i$ should be equal to their upper bound n_{mj}, since we cannot have "holes" in the curve, and all segments j should be "filled").
- n_s^m is the number of s-stop flights.

If f is the frequency-demand function, the previous example can be formulated as follows:

$$D_0^m = f(K_0^m n_0^m) \qquad (29)$$

$$D_1^m = D^m - D_0^m = f(K_0^m n_0^m + K_1^m n_1^m) - f(K_0^m n_0^m) \qquad (30)$$

If we had a general case, the curve would look like:

Demand

D_s^m

Frequency
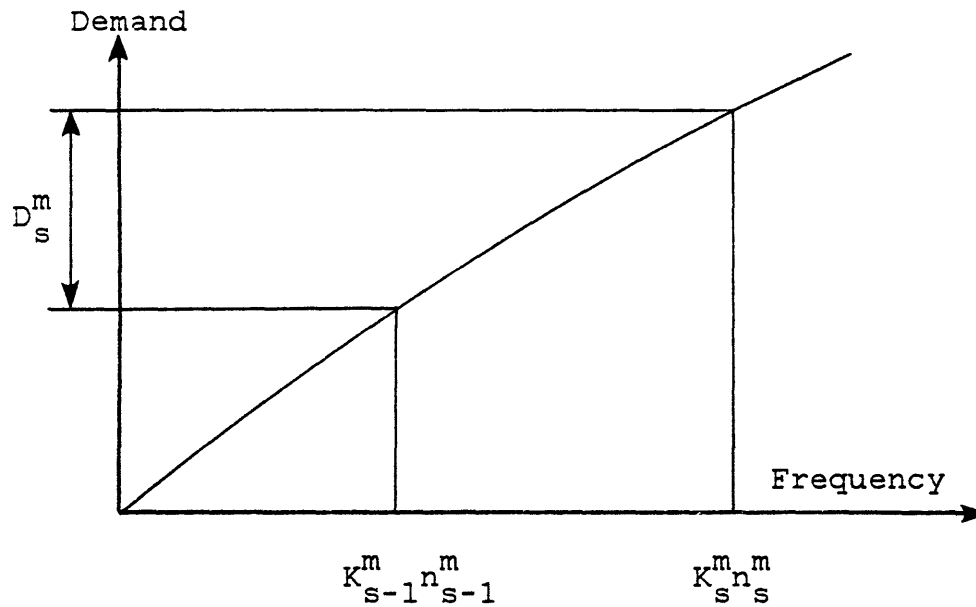
$K_{s-1}^m n_{s-1}^m$ $K_s^m n_s^m$

and the formulation would be:

$$D_s^m = f(K_0^m n_0^m + K_1^m n_1^m + \ldots + K_s^m n_s^m) - f(K_0^m n_0^m + \ldots + K_{s-1}^m n_{s-1}^m) \quad (31)$$

total frequency of flights   flights with
with s stops or less   s-1 stops or less

The modeling does not increase the number of data needed since only one curve is necessary. The only difference with before is how the demand is distributed along that curve.

11.3 *THE EQUATIONS IN FA4*

Another solution to the phantom frequency problem   90

11.3.1 *TRADITIONAL FA4 EQUATIONS*

Let us see how we can enter that demand into the FA4 equations. For this, we need first to review a part of the formulation which was presented in part one.

1. The traffic-frequency formulation .

   The curve is defined with the following notations:
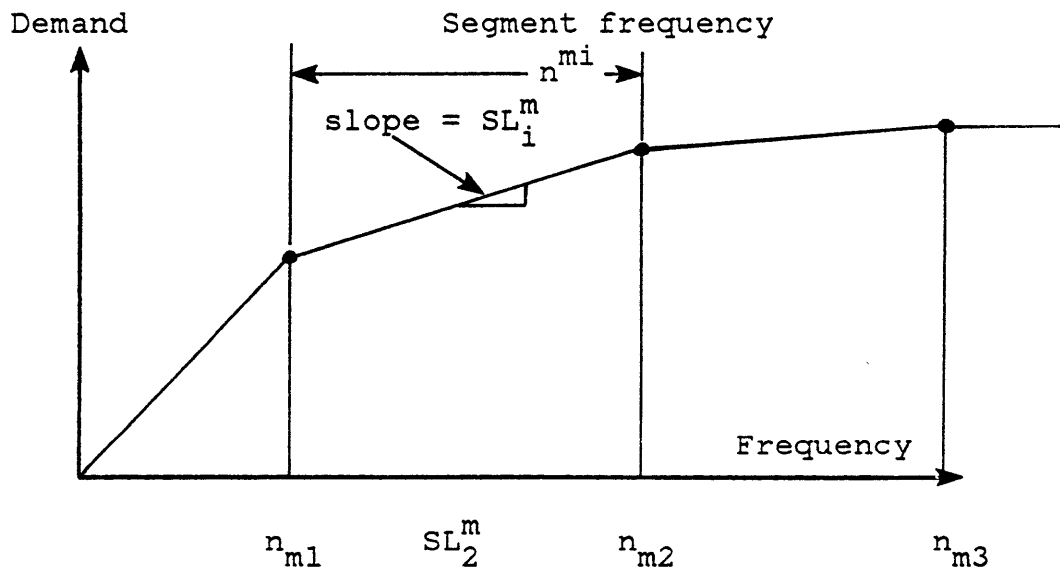
where n^{mi} is the ith break point frequency and P_{mi} the slope

of the ith curve segment. The first FA4 equation reviewed here represents the curve, it says that the total frequency

on a segment, n^{mi} is bounded:

$$0 \leq n^{mi} \leq n_{mi} - n_{m(i-1)} \qquad (32)$$

for all markets m, for all segments i of the curves.

with the previous curve it would mean:

$$0 \leq n^{m1} \leq n_{m1} - 0 \qquad (33)$$

$$0 \leq n^{m2} \leq n_{m2} - n_{m1} \qquad (34)$$

$$0 \leq n^{m3} \leq n_{m3} - n_{m2} \qquad (35)$$

the total frequency in the market m will be:

$$N^m = n^{mi} + n^{m2} + n^{m3} \qquad (36)$$

2. Second constraint in Traffic-Frequency formulation

Since we know the slopes of the segments: SL_i^m, the corresponding total demand can be computed:

$$D^m = f(N^m) = SL_1^m n^m + SL_2^m n^m + SL_3^m n^m \qquad (37)$$

This leads to another FA4 equation: The traffic carried on market m is a function of the total daily frequency:

$$\sum_i SL_i^m n^{mi} - \sum_{\underline{R}^m} T_r^m \geq 0 \qquad (38)$$

for all markets m.

Here it would be:

$$SL_1^m n^{m1} + SL_2^m n^{m2} + SL_3^m n^{m3} - \sum_{\underline{R}^m} T_r^m \geq 0 \qquad (39)$$

3. Third constraint - Traffic frequency curve

It says that:

the total daily frequency is the sum of the service frequencies.

$$\sum_s \sum_v K_{vs}^m \sum_{\underline{R}_s^m} n_{vr} - \sum_i n^{mi} = 0 \tag{40}$$

for all markets m.

which can be translated in our case example:

$$\sum_v K_{v0}^m \sum_{\underline{R}_0^m} n_{vr} + \sum_v K_{v1}^m \sum_{\underline{R}_1^m} n_{vr} - \sum_i n^{mi} = 0 \tag{41}$$

4. The segment load-factor constraint

$$\sum_v MAX\ LF_v^l\ S_v\ n_{vr} - \sum_m \underline{M}_r^l T_r^m \geq 0 \tag{42}$$

for all links l of routes r.

11.3.2 *THE NEW SET OF EQUATION*

Let us now introduce simply the D_s^m we defined at the beginning:

For non-stop flights: The traffic carried for market m non-stop is:

$$\sum_i SL_i^m \, n_0^{mi} - \sum_{\underline{R}_0^m} T_r^m \geq 0 \tag{43}$$

where the previous
$$n_0^m = \sum_i n_0^{mi} \tag{44}$$

(non-stop frequency)

and the total non-stop frequency is defined as follows:

$$\sum_v K_{v0}^m \sum_{\underline{R}_0^m} n_{vr} - \sum_i n_0^{mi} = 0 \tag{45}$$

For one-stop flights:

$$\begin{array}{l} \sum_i SL_i^m \, n_1^{mi} \\ - \sum_i SL_i^m \, n_0^{mi} \end{array} \quad - \sum_{\underline{R}_1^m} T^m \geq 0 \tag{46}$$

This is the equivalent in   This is the traffic on
simplified notations of: market m with one-stop
                                      routes

$$f(K_0^m n_0^m + K_1^m n_1^m) - f(K_0^m n_0^m) \tag{47}$$

and:

$$\sum_v K_{v0}^m \sum_{\underline{R}_0^m} n_{vr} + \sum_v K_{v1}^m \sum_{\underline{R}_1^m} n_{vr} - \sum_i n_1^{mi} = 0 \tag{48}$$

or in simplified formulation:

$$K_0^m n_0^m + K_1^m n_1^m = \sum_i n_1^{mi} \tag{49}$$

(total frequency of flights with one stop or less)

for these two cases, we must also define the bounds of the n_s^{mi}.
The next figure explains how the new frequency segments vari-
able are related to the curve:

Demand

total
frequency
for:

non-stop	n_0^{m1}	+	n_0^{m2}	+	n_0^{m3}	
1-stop or less	n_1^{m1}	+	n_1^{m2}	+	n_1^{m3}	
2-stops or less	n_2^{m1}	+	n_2^{m2}	+	n_2^{m3}	

n_{m1} $\quad$ n_{m2} $\quad$ n_{m3}
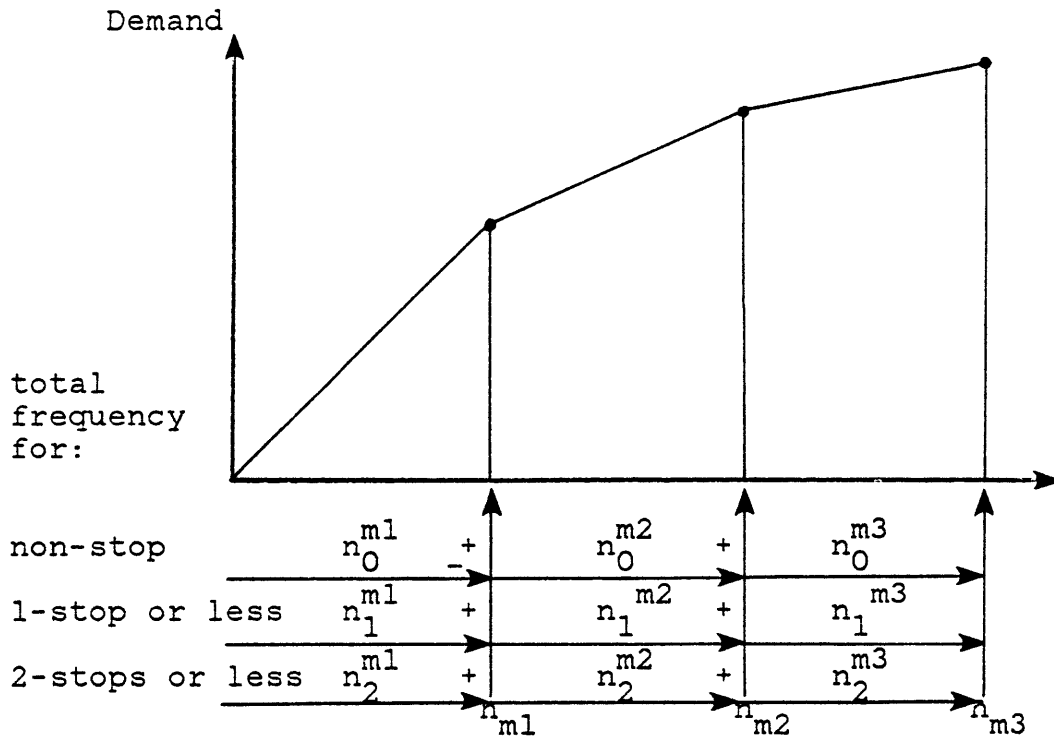
Since the curve is unique, the bounds of the n_s are the same for a given market:

$$0 \leq n_s^{mi} \leq n - n_{m(i-1)} \tag{50}$$

for all i, s and m.

Finally the load factor constraint is unchanged.

With this study, it is easy now to derive the general equations for s-stop flights. Only the equations which changed with respect to FA4 formulation, are presented:

$$\sum_i SL_i^m n_s^{mi} - \sum_{R_s^m} T_r^m \geq 0 \tag{51}$$
$$- \sum_i SL_i^m n_{s-1}^{mi}$$

$$\sum_v K_{v0}^m \sum_{R_0^m} n_{vr} + \ldots + \sum_v K_{vs}^m \sum_{R_s^m} n_{vr} - \sum_i n_s^{mi} = 0 \tag{52}$$

$$0 \leq n_s^{mi} \leq n_{mi} - n_{m(i-1)} \tag{53}$$

11.4 *AN EXAMPLE.*

Let us, finally, "analyse" a small example, in order to understand how this new modeling will fit into a general FA4 optimization.

1. The network is the same as the first defined with the cities A,B,C, D and X.

2. The routes are:

     BC

     ABC

     BCD

     BXC

     BXCD

3. The demand is:

   • high level of demand on markets AC,BD,BX,XC,XD

   • no demand on markets AB,CD

   • demand on market BC is distributed as follows:
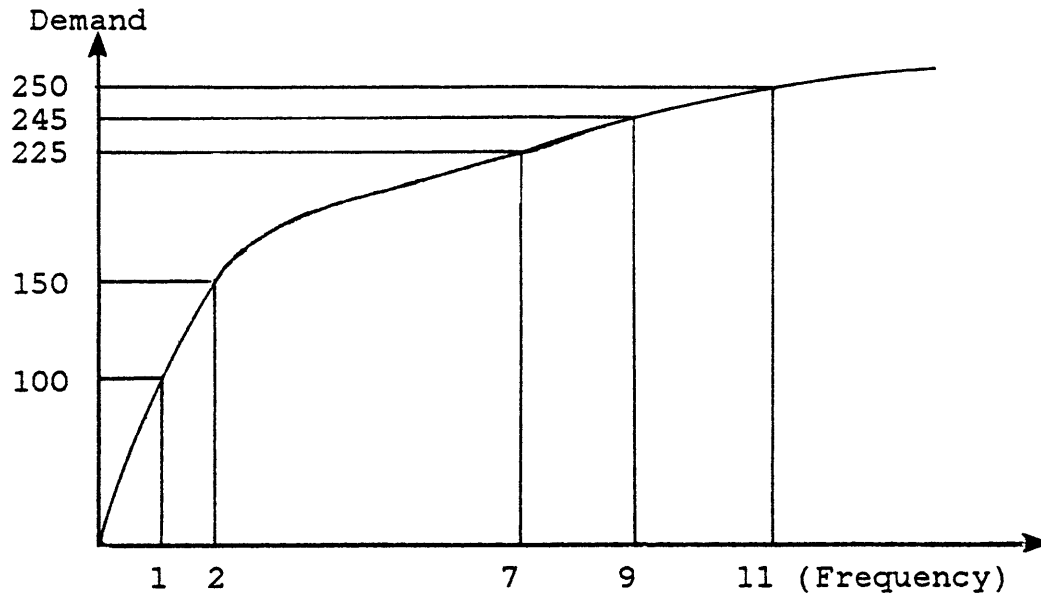
4. With the appropriate fares, landing fees etc... the results
   is supposed to be:

Route	Frequency	Aircraft capacity	Average load factor
BC	1 flight	200 seats	100%
ABC	3 flights	100 seats	100%
BCD	3 flights	110 seats	100%
BXC	4 flights	105 seats	100%
BXCD	4 flights	100 seats	100%

which is obtained as follows:

1. The equivalent frequency for market BC is:

   $1 + 3 + 3 + .5(4 + 4) = 11$

   (one-stop frequency weighting factor: $k_1 = 0.5$)

   which means:

   250 passengers which are split between routes BC, BCD, and

   BXC.

Another solution to the phantom frequency problem   97

and route BC is full with BC passengers: *200 passengers carried*

2. route ABC is full with AC passengers: *300 passengers carried*

3. route BCD has

   - BD passengers: *300 passengers carried*

   - BC passengers: *30 passengers carried*

4. route BXC has

   - BX passengers: *400 passengers carried*

   - XC passengers: *400 passengers carried*

   - BC passengers: *20 passengers carried*

5. route BXCD has

   - BX passengers: *400 passengers carried*

   - XD passengers: *400 passengers carried*

Just by looking at market BC traffic we can immediatly see the phantom frequencies:

- 3 on route ABC
- something less than 3 on route BCD: let say 2.5
- something less than .5(4)=2 on route BXC: let say 1.5
- 2 on route BXCD

This means that the equivalent frequency for market BC would only be:

$$1 + .5 + .5 = 2$$

which attracts 150 passengers approximately.

So we apply a " first treatment": separation of non-stop and one-stop flights (new solution suggested), which gives with the previous notations:

$$T_0^{BC} = T_{BC+ABC+BCD}^{BC} < f(7) = 225$$
$$T_1^{BC} = T_{BXC+BXCD}^{BC} < f(11) - f(7) = 25$$

where T_0^{BC} is the total traffic carried by non-stop flights on market BX on market BC,

T_1^{BC} the total traffic carried by one-stop flights ...

But here again, we see that the upper bounds on T_0^{BC} and T_1^{BC} are too high:

1. T_0 . Route BC will still carry 200 passengers and BCD the 25 remaining. This means that we still have phantom frequencies among non-stop flights for market BC.

2. T_1 . Since route BXCD does not carry BC passengers, its flights can be also considered as phantom frequencies.

We should then apply individually to each type of the flights the "old treatment" which was proposed in the past (FA6):

1. Non-stop flights:

$$T^{BC}_{BC} < f(1) \longrightarrow T^{BC}_{BC} = 100$$

$$T^{BC}_{ABC} < f(3)$$

$$T^{BC}_{BCD} < f(3)$$

$$T^{BC}_{BC + ABC} < f(4)$$

$$T^{BC}_{BC + BCD} < f(4) \longrightarrow T^{BC}_{BCD} = 30$$

$$T^{BC}_{ABC + BCD} < f(6)$$

$$T^{BC}_{BC + ABC + BCD} < f(7)$$

2. One-stop flights:

$$T^{BC}_{BXC} < f(9) - f(7) \longrightarrow T^{BC}_{BXC} = 20$$

$$T^{BC}_{BXCD} < f(9) - f(7)$$

$$T^{BC}_{BXC + BXCD} < f(11) - f(7)$$
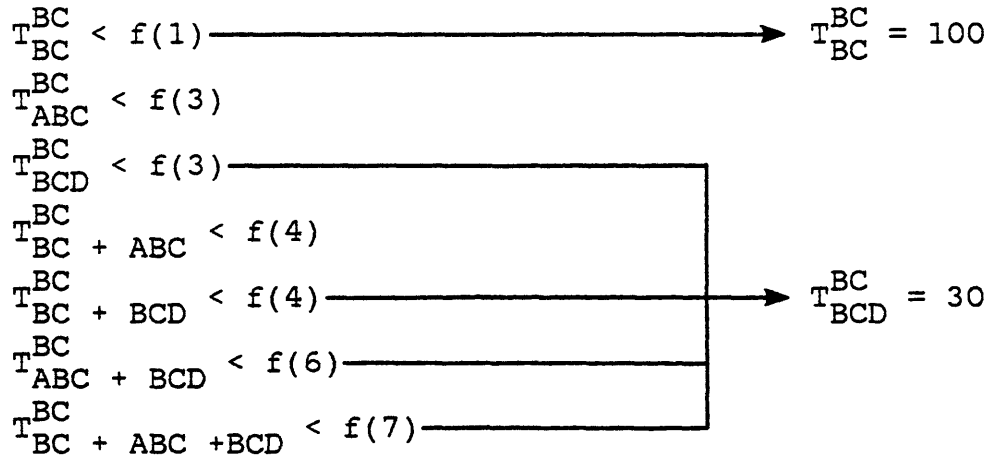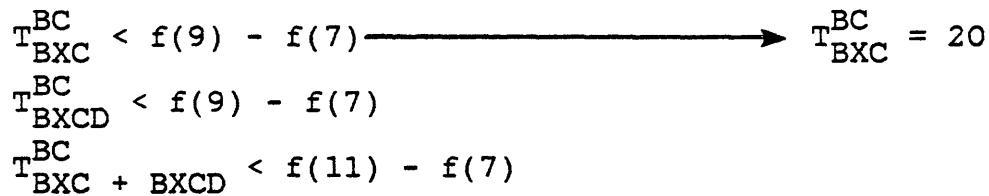
which gives a total of 150 passengers for market BC, and not 250.

11.5 DISCUSSION OF THE NEW MODEL

We have now a different treatment for different types of flights. As we can see, the way demand is now distributed, avoids phantom frequencies problems between flights which have a different number of stops. Within these flight categories, it is possible (as we have seen in the previous example) to use the answer proposed in FA6, but applied now to a smaller number of equations.

Let us see what the difference can be, by choosing a set of routes where we count flights up to 2 stops (MAX s = 2):

AB

ABC

ABD

AXBC

AXBD

AXYBC

AXYBD

Market AB is served by seven routes, which means, with the old solution of the phantom frequency problem:

$$c_1^7 + c_2^7 + c_3^7 + c_4^7 + c_5^7 + c_6^7 + c_7^7 = 127 \text{ new equations}$$

introduced to avoid phantom frequencies.

With the new proposed solution, we divide these routes into 3 "type-of-flight categories" (non-stop, one-stop ...), and inside each of them, we apply the "old treatment" to the new equations which were introduced previously. This means here:

$$(c_1^3 + c_2^3 + c_3^3) + (c_1^2 + c_2^2) + (c_1^2 + c_2^2) = 13 \text{ equations}$$

to which we have to add:

1. the definition of the n_s .

    If the frequency-demand curves have 4 segments, it means 8

    more

    equations.

2. 2 equations for the traffic carried.

3. 2 equations for the total equivalent frequencies.

Finally, the n^{mi} have been replaced by n^{mi}, n^{mi}, n^{mi},
which mean (MAX s) i = 2 i new variables. Instead of 127
equations, we have 21 of them plus 6 variables.

On the average, the new proposed solution should decrease sub-
stantially the number of equations needed. Of course, if we
have only one type of flight serving a given market, we are
reduced to the "old treatment". On a complete network,
however, this situation is rare.

The validity of the frequency weighting factors should also be
discussed here. These factors are computed, as we have seen,
from real market data and they are fixed at the same value what-
ever our new frequencies are. It can be wrong to do so, because
the optimization results might affect the re-partition of
flights on the market, and, thus, change the value of the fre-
quency weighting factors.

For example, if in real life a market is served by 5 non-stop frequencies, 4 of them being our flights, the weight of multi-stop flights will be fixed at a given value. After running FA4, we can imagine that, for economic and fleet availability reasons, the new result shows that our airline should fly only one non-stop service and a few one-stop flights. Since the competitors frequencies are assumed to be fixed, the global market is served now by only 2 non-stop frequencies instead of 5. The frequency weighting factors of multi-stop flights should, then, be changed.

A possible solution would be to rerun the frequency-demand curve generator with the new market caracteristics and rerun FA4 with the new curves. Will the solution converge ? The answer will be find with further research. But this problem raises the issue of wether the Fleet Assignment procedure can be a one time optimization or not, depending on the modeling proposed.

In the end, one might object: what if the s-stop services provided cannot carry the demand for s-stop flights?

The answer is simple: in the second modified equation the term in

$$- \sum_i SL_i^m \, n_{s-1}^m \qquad (54)$$

which represents all the *demand* generated by flights up to (s-1) stops can be replaced by the passengers *effectively* carried by these flights:

$$- \sum_{R_0^m} T_r^m - \sum_{R_1^m} T_r^m - \ldots - \sum_{R_{s-1}^m} T_r^m \qquad (55)$$

Only the formulation is changed.

Which one should be used ? This will be decided through future expriments, now possible with the interactive FA4.

BIBLIOGRAPHY

1. BARKLEY R.
   An optimum solution to an Aircraft Routing problem with multiple departure times. S.M. Thesis, MIT, June 1968

2. BENNETT J.E.
   Gravity model based on perceived travel-times. Transportation Engineering Journal, Vol 106 No TE1, January 1980 p.59

3. DANTZIG G.B., FERGUSON A.R.
   The problem of Routing Aircraft. A mathematical solution. Aeronautical Engineering Review, April 1955 p.51

4. DANTZIG G.B., FERGUSON A.R.
   The allocation of aircraft to routes. An example of linear programming under uncertain demand. Princeton University Press, Princeton N.J., 1963 p.568

5. ERIKSEN S.E.
   Demand models for US. Domestic Air Passenger markets. MIT, FTL Report R 78-2, June 1978

6. GAGNON G.
   A model for flowing passengers over Airline networks. Transportation Science I, 1967 p.232

7. GAINES B.R.
   Programming Interactive Dialogue. Proceedings of the ONLINE conference, Pragmatic Programming and Sensible Software, London, February 1978 p.306

8. GAINES B.R.
   On the complexity of causal models. IEEE, Transactions on systems, Man and Cybernetics, SMC-6(1), Jan 1976 p.56

9. GAINES B.R., FACES P.V.
   Some experience in interactive system development and applications. Proceedings IEEE 63, 1975 p.155

10. GUNN W.A.
    Airline System Simulation. Operation Research 12, 1964 p.206

11. HANDLER G.
    Assignment. Resolution of the Phantom Frequency problem FA6. MIT, FTL Memorandum M 74-5, February 1974

12.   HELDT R.W.
A solution algorithm for the Aircraft Routing problem with multiple departure times. S.M. Thesis, MIT, June 1969

13.   HOWARD L.R., EBERHARDT D.O.
Airline Simulation for analysis of Commercial Airplane markets. Transportation Science, vol 1 No 3, August 1967 p.131

14.   KUSHIGE, TAKAYASU
A solution of most profitable Airline Routing (An application of linear programming). Japan Air Lines, Proceedings of the third AGIFORS Symposium, Canada, Oct 1973 p.157

15.   LEVIN A.
Some Fleet Routing and Scheduling problems for Air Transportation systems. MIT, ORSA Third annual Israel conference on Operation Research, July 1969

16.   MARTIN J.
Design of man-computer dialogue. Prentice Hall, New Jersey, 1973

17.   MARSTEN R.E., MULLER M.R.
Network design and Aircraft Fleet Planning at Flying Tiger: a succesful application of mixed-integer programming. Flying Tiger, University of Arizona, January 1980

18.   MATHAISEL D.F.X.
An Airline Schedule Construction model. S.M. Thesis, University of California Irvine, May 1975

19.   MATHAISEL D.F.X.
A long range Fleet Planning Model for the Airline Industry. MIT, FTL Memorandum M 77-6, May 1977

20.   MATHAISEL D.F.X.
Fleet Planning for the Air Transport Industry. Network Aggregation and Cell Theory models. Ph.D. Thesis, MIT, December 1980

21.   MATHAISEL D.F.X.
Quantifying Airline Scheduling behavior for use in determining the relationship between passenger displacement time and frequency of service. MIT, FTL, January 1977

22.   POLLACK M.
Some elements of the Airline Fleet Planning problem. Transportation Research Vol 11 No 5, 1977 p.301

23.   POLLACK M.

An interactive approach to Airline Route-Frequency planning. INFORMATION processing 74, North Holland Publishing Company (1974)

24.   POLLACK M.
An interactive computer model for Airline Frequency Planning. Looked California Company, Report CTR/1660, October 1973

25.   ROBINSON A.C.
Optimal Scheduling and Routing in Air Transportation networks. Battell Memorial Institute, ASME Report No.70 TRAN-37, July 1970

26.   SIMPSON R.W.
Scheduling and Routing models for Airline systems. MIT, Flight Transportation Laboratory (FTL) report R68-3, Dec 1969

27.   SIMPSON R.W.
A theory for Domestic Airline Economics. MIT, FTL Report R 74-2, July 1974

28.   SIMPSON R.W.
A market-share multi-stop Fleet Assignment model FA4. MIT, FTL Memorandum M 70-7, June 1970

29.   SIMPSON R.W., HANDLER G.
A Fleet Assignment model incorporating connecting services FA5. MIT, FTL Memorandum M 74-4, February 1974

30.   SIMPSON R.W., SWAN W.M.
Network models for Transportation Policy Analysis. D.O.T. Report DOT.TST.77.81, January 1977

31.   SWAN W.M.
FA7: 1977 version of Linear Programming Fleet Assignment model. MIT, FTL, Spring 1977

32.   SWAN W.M.
Description of input for FA4. MIT, FTL Memorandum M 72-4, February 1972

33.   SWAN W.M.
The complete FA4 Memo. MIT, FTL Memorandum M 72-10, August 1972

34.   SWAN W.M.
FA4.5 Users Memo. MIT, FTL Memorandum M 73-3, January 1973

35.   SWAN W.M.

Explanation, validity and use of Modal-Split models to generate Traffic-Frequency curves for FA4. MIT, FTL Memorandum M 73-7, May 1973

36.   TARTER J.
       Air Florida. The little airline that could.   INC., January 1981 p.56

37.   *User's Manual for FA4. Airline Fleet Assignment and Planning Model.*
       MIT, FTL, September 1976

APPENDICES

A.1 INTRODUCTION

 IFA/1 is an interactive tool designed to create and easily
modify the data decks needed by the fleet assignment model FA4.
Before working with IFA/1 the user should be familiar with the
FA4 options and the data required to run FA4.

 Throughout this manual, the syntax of the commands has the
following convention.
 Each command has a long and a short form. The long form is
always written with the letters corresponding to the short form
in upper case.
 Example:
    the command CYCLEFRQ can also be written CF
    its syntax will then be written: CycleFrq <new value>

 Finally, in the examples given in the manual:
 Each time the user has to enter a line, it is preceded by an
arrow: ->
 When lines of the real output are skipped the sign (...) is
used.

A.2 *THE INTERACTIVE COMMANDS*

IFA1 is organized with 3 levels of commands. In the first two levels, the user will enter the command after specific "command prompts". Level 3 commands, on the other end, will be entered as options of level two commands.

A.2.1 *LEVEL 1*

The user can choose the data deck to be processed and whether it should be created, modified or saved.

The specific "command prompt" is: 1.>

The commands available are:

```
<BUILD> or <B>    : to create a new set of data.
 syntax:   Build <data deck>

<MODIFY> or <M>  : to modify an existing set of data.
 syntax:  Modify <data deck>

<SAVE> or <S>     : to save (into a disk file) a processed
                    data set.
 syntax:  Save <data deck>    ->  saves one data deck.
          Save                ->  saves all the data decks
                                  previously created or
                                  modified.

<END>             : to end the session and leave the
                    interactive environment.
 syntax:  END
```

In response to this command, the following message is displayed:

        HAVE YOU SAVED YOUR FILES ? (ENTER: Y OR SAVE)

The user should then type:

        Y   (for yes) if the data have been saved already.
        SAVE to save all the data decks which were created
              or modified and terminate the session.

The DATA DECKS are:

```
<SYSTEM>   or <ST>     for general system data.

<AIRCRAFT> or <AC>     for aircraft data.

<AIRPORT>  or <AP>     for airport data.

<CITYPAIR> or <CP>     for city-pair data.

<ROUTE>    or <RT>     for route data.
```

EXAMPLE:
The user wants to create an airport data deck, save it and
leave IFA1.

```
    1.>
->  build airport
    THIS COMMAND WILL ERASE THE PREVIOUS AIRPORT DATA SET
    DO YOU WANT TO CONTINUE ? (ENTER Y OR N)
->  y
    AIRPORT NAME (3 LETTER CODE),AIRPORT ACTIVITY LEVEL?
->  aaa 23
 (...)
    DATA OK? (ENTER: Y,N,B OR C)
->  y
    MORE AIRPORTS? (ENTER: Y OR N)
->  n
         25 AIRPORT(S) IN THE LIST
    1.>
->  save airport
    FILE AP SAVED WITH     25 AIRPORT(S)
    1.>
->  end
    HAVE YOU SAVED YOUR FILES ? (ENTER: Y OR SAVE)
->  y
    SESSION TERMINATED AT ...
```

A.2.2 *LEVEL 2*

1. From the BUILD command
   The user will be prompted for specific data to be entered
   into the corresponding deck. For example:

   There are 3 ways of answering these prompts:

   •   just enter the data, separated by a space or a comma.
       The prompt for the next data will then appear.

   •   press the ENTER or RETURN key and the next data prompt
       will appear. The previous data will remain unchanged.

- enter the letter B (for back) and the previous data prompt will be displayed.

This allows the user to go through the data prompts, forewards and backwards, in order to correct eventual mistakes. After the last data prompt the data entered will be displayed as they will be stored, and the following message will appear:

DATA OK ? (ENTER: Y,N,B OR C)

The user should type:

Y   (for yes) if the data has been properly entered. The user will then return to level 1.

N   (for no) if the data is not correct. The user goes back to the first data prompt, but as long as they are not modified, the data previously entered remain unchanged.

B   (for back) the last data prompt is displayed again.

C   (for cancel) the BUILD session is cancelled and the data rejected. The user will return at level 1. (This can be used, for instance, when the data are not correct but there is no time to change them again.)

EXAMPLE:
 We are in the airport data deck
  AIRPORT NAME (3 LETTER CODE),AIRPORT ACTIVITY LEVEL?
->
  LANDING FEES BY AIRCRAFT TYPE?
->b
  AIRPORT NAME (3 LETTER CODE),AIRPORT ACTIVITY LEVEL?
->aaa 344
  LANDING FEES BY AIRCRAFT TYPE?
->1234 200 123 500
 (...)
  DATA OK? (ENTER: Y,N,B OR C)
->b
  AIRPORT NAME ...
 (...)
  DATA OK? (ENTER: Y,N,B OR C)
->c
  1.>
 (...)

2. From the MODIFY command

Depending on which data deck has been selected, the "command prompt" and the commands available are different.

a. SYSTEM data

The prompt is ST.>

The commands available are:

<CHANGE> or <C>   : to modify a given data.
 syntax: Change   <level 3 commands>

<PRINT> or <P>   : to display the whole data
                       deck.
 syntax: Print

b. AIRCRAFT data

The prompt is AC.>

The commands available are:
<CHANGE> or <C>   : to change a given data of a
                       given aircraft.
 syntax: Change <aircraft name><level 3 commands>
The corresponding aircraft data is displayed
afterwards for a final check.

<ADD> or <A>   : to add a new aircraft to the
                       data deck.
 syntax: Add
The same data prompts as in the BUILD command
will appear and the procedure remains identical.

<DELETE> or <D>   : to delete an aircraft and the
                       corresponding data from the
                       deck.
 syntax: Delete   <aircraft name>

<PRINT> or <P>   : again to display data.
 syntax: Print <aircraft name> -> displays one
                                       aircraft data.
            Print   -> displays the
                                 whole data deck.

The <aircraft name> is the code name given in the
data deck.

c. AIRPORT data

The prompt is AP.>

The commands available are:

```
<CHANGE> or <C>      : same as before (see aircraft)
  syntax:   Change <airport name><level 3 commands>

<ADD> or <A>         : same as before (see aircraft)
  syntax:   Add

<DELETE> or <D>      : same as before (see aircraft)
  syntax:   Delete  <airport name>

<PRINT> or <P>       : same as before (see aircraft)
  syntax:   Print  <airport name>
            Print
```

The <airport name> is the three letter code name given in the data deck.

d. CITY-PAIR data

   The prompt is   CP.>

   The commands available are:

```
<CHANGE> or <C>      : same as before (see aircraft)
  syntax: Change<city-pair name><level 3 commands>

<ADD> or <A>         : same as before (see aircraft)
  syntax:   Add

<DELETE> or <D>      : same as before (see aircraft)
  syntax:   Delete  <city-pair name>

<PRINT> or <P>       : same as before (see aircraft)
  syntax:   Print  <city-pair name>
            Print
```
<city-pair name> consist of the <name of the first city> and <name of the second city> both written with the three letter code as in the deck.

e. ROUTE data

   The prompt is   RT.>

   The commands available are:

```
<CHANGE> or <C>      : same as before (see aircraft)
  syntax: Change <route number> <level 3 commands>

<ADD> or <A>         : same as before (see aircraft)
  syntax:   Add
```

```
        <DELETE> or <D>     : same as before (see aircraft)
          syntax:    Delete   <route number>


        <PRINT> or <P>      : same as before (see aircraft)
          syntax:    Print   <route number>
                     Print
```

The <route number> is the same as entered in the data deck.

 After the execution of these commands the system goes back to level 2 with the specific "command prompt". In order to go back to level 1 the user just presses the ENTER or RETURN key.

EXAMPLE:

```
 We are at level one and we want to change AIRPORT data
 and return to level 1
      1.>
->   modify airport
          25 AIRPORT(S)  IN THE LIST
      AP.>
->   change aaa up_bnd 600
      AAA    600. 1234.  120.  500.  100.
      AP.>
->   print aaa
      AAA    600. 1234.  120.  500.  100.
      AP.>
->
      1.>
  (...)
```


A.2.3 _LEVEL 3_

  As we have seen, level 3 commands are options of the level 2 CHANGE command. Their syntax is specific to the data set processed.

1. SYSTEM data

```
    <TITLE> or <TI>      : to change the title of the run.
      syntax:  TItle   <text of the new title>


    <DIL_FACT> or <DF>  : changes the dilution factor.
      syntax:  Dil_Fact   <new value>


    <FIX_FARE> or <FF>  : fixed part of the fare
      syntax:  Fix_Fare   <new value>
```

```
<FARE_1> or <F1>      : per-mile part of the fare
                        (mileage < 500 miles)
 syntax:  Fare_1  <new value>

<FARE_2> or <F2>      : same as before for
                        500 mi < mileage < 1500 mi

<FARE_3> or <F3>      : same as before for
                        1500 mi < mileage

<MAX_LF> or <ML>      : maximum load-factor multiplier
 syntax:  Max_Lf  <new value>

<LAND_FEE> or <LF>    : landing fee per aircraft weight
 syntax:  Land_Fee  <new value>
```

2. AIRCRAFT data

```
<SEAT_NUM> or <SN>    : to change the number of seats.
 syntax:  Seat_Num  <new value>

<AC_NUM> or <AN>      : changes the number of aircraft
                        available.
 syntax:  Ac_Num  <new value>

<UTILZTN> or <UT>     : aircraft utilization
                        (see FA4 manual for the options)
 syntax:  UTilstn  <new value>

<FUEL_PRC> or <FP>    : price of one unit of fuel.
 syntax:  Fuel_Prc  <new value>

<COST/HR> or <CH>     : aircraft direct cost per hour.
 syntax:  Cost/Hour  <new value>

<IOC/DEP> or <ID>     : aircraft indirect cost per
                        departure
 syntax:  Ioc/Dep  <new value>

<DESIRAB> or <DS>     : aircraft desirability
 syntax:  DeSirab  <new value>
```

3. AIRPORT data

```
<UP_BND> or <UB>      : maximum activity (upper bound).
 syntax:  Up_Bnd  <new value>

<LO_BND> or <LB>      : minimum activity (lower bound).
 syntax:  Lo_Bnd  <new value>
```

```
<LANDFEE> or <LF>   : aircraft landing-fee
 syntax:  UTilstn  <aircraft rank> <new value>
This command allows the user to change the landing-fee
for a given aircraft. This aircraft is indicated by
its rank in the aircraft data deck.
```

4. CITY-PAIR data

```
<MOD_PT> or <MP>      : to modify a point on the
                         frequency-demand curve.
 syntax:
 Mod_Pt <point number> <new frequency> <new demand>

<ADD_PT> or <AP>      : to add a new point to the
                         frequency-demand curve.
 syntax:   Add_Pt  <new frequency> <new demand>

<DEL_PT> or <DP>      : to delete a point from the
                         frequency-demand curve.
 syntax:  Del_Pt  <point number>

<LEV_SERV> or <LS>  : change the desired optional
                       level of service definition.
                       (see FA4 users manual)
 syntax:
 Lev_Serv <service-type index><frequency><flight-type>

<MAX_LF> or <ML>     : maximum allowable load-factor.
 syntax:  Max_Lf  <new value>
```

5. ROUTE data

```
<ADD_CITY> or <AC>  : add a new city in the route.
 syntax:
 Add_City <new city code> <code of the following city>
 Add_City <new city code> if the new city is to be the
                          last one of the route.

<DEL_CITY> or <DC>  : delete a city from the route
 syntax:  Del_City  <city code>

<ACNUMBER> or <AN>  : forced aircraft rank.
                      (see FA4 manual for the option)
 syntax:  AcNumber  <aircraft rank>

<CYCLEFRQ> or <CF>  : corresponding forced frequency
                      per cycle (see FA4 manual).
 syntax:  CycleFrq  <new value>
```

```
        <LBND> or <LB>        : logical variable to enable or
                                not the forcing option.
          syntax:   LBnd   <new value>


EXAMPLE:

We want to add a new city in a route
     1.>
->  modify route
         240 ROUTE(S) IN THE LIST
     RT.>
->  print 28
     28   0   70   AAA BBB CCC DDD
     RT.>
->  change 28 add_city fff ccc
     28   0   70   AAA BBB FFF CCC DDD
     RT.>

We want to write a new title
     1.>
->  modify system
     ST.>
->  change title this is a new title
     THIS IS A NEW TITLE
     DAY   YEAR   365. 28 ...
     ST.>

We want to change the city-pair optional level of service
     1.>
->  modify citypair
         1200 CITY-PAIR(S) IN THE LIST
     CP.>
->  change aaa bbb lev_serv 1 25 5
     AAA BBB   ...
     CP.>
  (...)
```

A.3 SUMMARY OF THE COMMANDS AND DATA DECKS AVAILABLE

```
level 1                     DATA decks
<BUILD>   <B>                  <ST>
<MODIFY>  <M>               <AIRCRAFT>  <AC>
<SAVE>    <S>               <AIRPORT>   <AP>
<END>                       <CITYPAIR>  <CP>
                            <ROUTE>     <RT>
-SYSTEM data-
level 2           level 3
<CHANGE> <C>     <TITLE>       <TI>
<PRINT>  <P>     <DIL_FACT>    <DF>
                 <FIX_FARE>    <FF>
                 <FARE_1>      <F1>
                 <FARE_2>      <F2>
                 <FARE_3>      <F3>
                 <MAX_LF>      <ML>
                 <LAND_FEE>    <LF>


   -AIRCRAFT data-
   level 2           level 3
   <CHANGE> <C>     <SEAT_NUM> <SN>
   <ADD>    <A>     <AC_NUM>   <AN>
   <DELETE> <D>     <UTILZTN>  <UT>
   <PRINT>  <P>     <FUEL_PRC> <FP>
                    <COST/HR>  <CH>
                    <IOC/DEP>  <ID>
                    <DESIRAB>  <DS>
      -AIRPORT data-
      level 2           level 3
      <CHANGE> <C>     <UP_BND>    <UB>
      <ADD>    <A>     <LO_BND>    <LB>
      <DELETE> <D>     <LANDFEE>   <LF>
      <PRINT>  <P>


         -CITY-PAIR data-
         level 2           level 3
         <CHANGE> <C>     <MOD_PT>   <MP>
         <ADD>    <A>     <ADD_PT>   <AP>
         <DELETE> <D>     <DEL_PT>   <DP>
         <PRINT>  <P>     <LEV_SERV> <LS>
                          <MAX_LF>   <ML>
            -ROUTE data-
            level 2           level 3
            <CHANGE> <C>     <ADD_CITY> <AC>
            <ADD>    <A>     <DEL_CITY> <DC>
            <DELETE> <D>     <ACNUMBER> <AN>
            <PRINT>  <P>     <CYCLEFRQ> <CF>
                             <LBND>     <LB>
```

r * is a route, a sequence of cities visited by an aircraft on one flight.

m * is a market, a pair of cities in which air transport service is available.

l * is a link which joins a pair of cities if it is used as a route segment.

s * is the number of stops on a route.

h, j * are used to represent airports.
j is a given airport, h represents the "other airports".

i * is the rank of the segments in the demand curve.

v * refers to a particular aircraft type (vehicle type).

R^m * is the subset of routes serving market m.

R^m_s * is the subset of routes serving market m with s stops.

R^{hj} * is the subset of routes serving airport j from airport h.

M^l_r * is the subset of markets which are served by link l of route r.

y^m * is the yield (ticket income) for a passenger travelling on market m.

T^m_r * is the number of passengers (traffic) on market m, route r.

VAC_{vr} * is the variable operating cost for aircraft v on route r.

n_{vr} * is the number of flights for aircraft v on route r.

n^{mi} * is the frequency we have reached on segment i of the demand curve for market m.

n_{mi} * is the upper frequency for segment i of the demand curve for market m.

n_s^m * is the number of s-stop flights on market m.

MAX LF_v^l * is the maximum desirable average load-factor for aircraft type v on link l.

S_v * is the number of seats on aircraft type v.

K_{vs}^m * is the frequency weighting factor for an s-stop flight on market m with aircraft type v.

SL_i^m * is the slope of segment i of the demand curve for market m.

BT_{vr} * is the average block-time required for aircraft type v to fly route r (not including gate ground times, but including all flight segments of route r).

A_v * is the available number of active aircraft of type v.

MAX U_v * is the maximum utilisation in average block-hours per available aircraft per schedule period.

MIN n^m * is the minimum desired frequency on market m.

MAX N^j * is the maximum number of departure from airport j.

VIC_a * is the variable indirect costs for our airline a.

Definition of variables in FA4   122

```
TEST CASE WITH MIXED FLEET
 &SCALES CYCLE='DAY', CYCLE2='YEAR', XOUT=365., LFUEL=.FALSE.,
    FUELNM='GAL', FULMIT=0., LSKED=.FALSE., &END
 &IOCS CBASIC=1000., CRPM=.006, CACMIL=.3, CBOARD=2.75, CDEPAR=20.,
    LPOLCY=.FALSE., &END
 DC-9   100.    75.    7.5  0.234     0.   450.     0.   460.    20.     0.32000.
 US40   100.    40.    7.5  0.117     0.   430.     0.   325.    15.     0.2 700.

 &FARED LFARE=.TRUE., FARETO=11.3, FAREMI=.0687, YIELD=.8, &END
 &FRQD LXK=.FALSE., XK0=1., XK1=.5, XK2=.25, VARILF=0.5, &END
 ATL    0.     22.68      10.63
 BNA    0.     16.32       7.80
 HSV    0.     16.32       7.80
 IAD    0.     18.14       8.66
 CSG    0.     15.          7.

 ATL BNA            1.                214.          0. 0
            3  2    30.   4    45.  8    55.
 1.  .7  .5  1.  .7  .5
 ATL HSV            1.                151.          0. 0
            4  2    70.   4   110.  8   135. 15   150.
 1.  .5 .25  1.  .5 .25
 ATL CSG            1.                 83.          0. 0
            4  4    50.   8    80. 12   100. 20   120.
 1.  .5 .25  1.  .5 .25
 ATL IAD            1.                540.          0. 0
            4  4   175.   8   225. 12   250. 20   270.
 1.  0.  0.  1.  0.  0.
 IAD CSG            1.                617.          0. 0
            3  2    20.   4    30. 10    35.
 1.  .8  .5  1.  .8  .5
 IAD HSV            1.                604.          0. 0
            4  2    30.   4    50.  8    75. 15    95.
 1.  .8  .5  1.  .8  .5
 IAD BNA            1.                552.          0. 0
            3  2    20.   4    30. 10    35.
 1.  .8  .5  1.  .8  .5
 BNA HSV            1.                103.          0. 0
            3  2    15.   4    23.  8    30.
 1.  .5 .25  1.  .5 .25

 &RTESD ICOST=2, LBND=.FALSE., VARBND=1., &END
     1 IAD ATL
     2 IAD CSG
     3 IAD HSV
     4 IAD BNA
     5 ATL CSG
     6 ATL HSV
     7 ATL BNA
    21 BNA HSV ATL
    22 BNA HSV IAD
    23 HSV ATL IAD
    24 CSG ATL IAD
    26 BNA ATL IAD
    31 BNA HSV ATL IAD
```

D.0 *EXAMPLE OF FA4 POSTPROCESSOR OUTPUT*

```
****RUN 0 :NO CONSTRAINT APPLIED TO THE SYSTEM


1. ROUTE DATA
   *************

   THE VALUES ARE GIVEN ARE FOR ONE WAY TRAFFIC.

        ROUTE 1111     MIA  TO ATL  TO ORD
                   NO FLIGHTS ARE FLOWN ON THIS ROUTE

        ROUTE 1112     MIA  TO ATL  TO ORD  TO MSP
                   NO FLIGHTS ARE FLOWN ON THIS ROUTE

        ROUTE 1113     MIA  TO ATL  TO ORD  TO MKE
                 0.0  PASSENGERS FROM ORD TO MKE
                60.74 PASSENGERS FROM ATL TO MKE
                 0.0  PASSENGERS FROM MIA TO MKE
                 0.0  PASSENGERS FROM ATL TO ORD
                 0.0  PASSENGERS FROM MIA TO ORD
                60.74 PASSENGERS FROM MIA TO ATL
                 0.60 TRIPS PER WEEK
                 0.60 TRIPS USING 727S

2. ROUTE SEGMENT DATA
   ************************

   THE VALUES ARE GIVEN ARE FOR ONE WAY TRAFFIC.

      ROUTE SEGMENT  DTW TO ATL
                260.46 PASSENGERS.  LOAD FACTOR=0.65  MAXIMUM PERMISSIBLE LOAD FACTOR=0.65
                 43.79 PASSENGERS FROM ROUTE 1121
                 52.93 PASSENGERS FROM ROUTE 1123
                163.74 PASSENGERS FROM ROUTE 1125

      ROUTE SEGMENT  DTW TO DLH
                NO SERVICES FROM ANY ROUTES

      ROUTE SEGMENT  FSD TO DTW
                NO SERVICES FROM ANY ROUTES

3. CITY PAIR ORIGIN AND DESTINATION DATA
   *****************************************

   THE VALUES ARE GIVEN ARE FOR ONE WAY TRAFFIC.

      CITY PAIR  DTW TO ATL
             163.74 PASSENGERS.  2.59 SERVICES
                 163.74 PASSENGERS.  2.59 NONSTOP SERVICES FROM ROUTE(S) 1121. 1123. 1125

      CITY PAIR  DTW TO DLH
             NO SERVICE FOR THIS CITY PAIR

      CITY PAIR  FSD TO DTW
             NO SERVICE FOR THIS CITY PAIR
```

Example of FA4 postprocessor output   124

4. SYSTEM ECONOMICS IN DOLLARS PER YEAR
****

```
        REVENUE      =    71061216.0
        DOC          =    36742512.0
        IOC          =    24357344.0
        LANDING FEES =      835754.56
        CONTRIBUTION =     9125605.00
        OBJECTIVE FN =     9125038.00
```

5. SYSTEM TRAFFIC PER YEAR
****

```
        PASSENGERS   =      884444.812
            95.12% TRAVELED WITH 0 STOPS.  AVERAGE TRIP LENGTH=  887.39 MILES
             4.88% TRAVELED WITH 1 STOPS.  AVERAGE TRIP LENGTH= 1076.96 MILES
             0.0 % TRAVELED WITH 2 STOPS.  AVERAGE TRIP LENGTH=    0.0  MILES
             0.0 % TRAVELED WITH 3 STOPS.  AVERAGE TRIP LENGTH=    0.0  MILES
             0.0 % TRAVELED WITH 4 STOPS.  AVERAGE TRIP LENGTH=    0.0  MILES
             0.0 % TRAVELED WITH 5 STOPS.  AVERAGE TRIP LENGTH=    0.0  MILES

        RPM          =   793027584.
        RM           =     8323523.00
        TAKE OFFS    =       13614.937
        FLIGHTS      =        6956.109
        LF(PAX/SEATS)=           0.440
        LF(RPM/ASM)  =           0.615
```

6. FLEET USAGE DATA PER YEAR
****

AIRCRAFT	HRS ACTIVE	HRS IDLE	TAKEOFFS PER HR	AVERAGE STAGE LENGTH
727S	16785.51	420014.44	0.811	611.35
D910	0.0	436800.00	0.0	0.0
D930	0.0	436800.00	0.0	0.0
F-27	0.0	436800.00	0.0	0.0

AIRCRAFT NAME	*	FUEL CONSUMED IN GAL	*	FUEL EXPENSE IN $	*	TOTAL DOC IN $
727S	*	22839856.	*	11362828.0	*	36742512.
D910	*	0.	*	0.0	*	0.
D930	*	0.	*	0.0	*	0.
F-27	*	0.	*	0.0	*	0.
TOTALS :		22839856.		11362828.0		36742512.

Example of FA4 postprocessor output   125

```
          *** INTERACTIVE FLEET ASSIGNMENT - IFA1 ***
SESSION STARTED AT 14:19:53
1.>
m odify route
COMMAND M ODIFY NOT RECOGNIZED: ODIFY ?
1.>
modify route
          2 ROUTE(S) IN THE LIST
RT.>
R
.,  FALSE        1.
    1    1   70 AAA BBB RRR CCC
    1 12
    4    1   70 DDD BBB HHH RRR MMM

RT.>
change 1 ac 111 rrr
    1    1   70 AAA BBB LLL RRR CCC
    1 12
RT.>
change 4 dc hhh
    4    1   70 DDD BBB RRR MMM

RT.>
.
1.>
m ap
          6 AIRPORT(S) IN THE LIST
AP.>
R
AAA     23.        1.
BBB     23.       12.      12.      12.      12.
DDD     34.       23.    2323.      23.
EEE    234.       34.      34.      34.
FFF     23.       12.      12.
MMM     34.     1234.     345.    6789.
AP.>
a
AIRPORT NAME (3 LETTER CODE),AIRPORT ACTIVITY LEVEL?
sss 500
LANDING FEES BY AIRCRAFT TYPE?
123 123 123 123 123 123 123 123 123
NUMBER OF AIRCRAFT EXCEEDED
LANDING FEES BY AIRCRAFT TYPE?
123 123 123
GGG    500.     123.     123.     123.
DATA OK? (ENTER: Y,N,B OR C)
y
AP.>
```

```
1.>
b st
THIS COMMAND WILL ERASE THE PREVIOUS SYSTEM DATA SET
DO YOU WANT TO CONTINUE? (ENTER Y OR N)
y
TITLE OF THE RUN?

OPERATION CYCLE,SYSTEM OUTPUT CYCLE,SCALE FACTOR?
b
TITLE OF THE RUN?
this is a demonstration run
OPERATION CYCLE,SYSTEM OUTPUT CYCLE,SCALE FACTOR?
day year 365
MAXI FUEL CONSTRAINT LOGICAL,FUEL UNIT,FUEL AVAILABLE?
false gal 0
SKEDGEN OUTPUT LOGICAL,FIXED SYSTEM COST PER CYCLE?
false 1000
MARGINAL IOC/RPM,MARGINAL IOC/AM,MARGINAL COST PER PAX ENPLANED?
200 0 53
MARGINAL IOC/LAND & TAKEOFF CYCLE,OBJECTIVE FUNCTION LOGICAL?
234 false
FARE COMPUTATION LOGICAL,REVENUE DILUTION FACTOR,LAND.-FEE/WEIGHT?
b
MARGINAL IOC/LAND & TAKEOFF CYCLE,OBJECTIVE FUNCTION LOGICAL?
b
MARGINAL IOC/RPM,MARGINAL IOC/AM,MARGINAL COST PER PAX ENPLANED?

MARGINAL IOC/LAND & TAKEOFF CYCLE,OBJECTIVE FUNCTION LOGICAL?

FARE COMPUTATION LOGICAL,REVENUE DILUTION FACTOR,LAND.-FEE/WEIGHT?
false .95 1000
C.A.B. FARE FORMULA: FIXED PART,FARE/MILE (0->500,->1500 & >1500?
43 .750 .707 .689
FREQ. WEIGHTING FACTOR LOGICAL, MAXI LOAD-FACTOR MULTIPLIER?
true .74
FREQUENCY WEIGHTING FACTORS ?(COMPULSORY)
1 .9 .8 .7 .6 .5 .4 .3 .2 .1
MAXIMUM NUMBER OF FACTORS EXCEEDED
FREQUENCY WEIGHTING FACTORS ?(COMPULSORY)
1 .9 .5 .1
THIS IS A DEMONSTRATION RUN
     DAY    YEAR    365,FALSEGAL        0,FALSE 1000,    200,      0,
FALSEFALSE    .95 1000,     43,  .750  .707  .689 TRUE     .74
     1,    .9    .5     .1
DATA OK? (ENTER: Y,N,B OR C)
y
1.>
```

```
m cp
- -  ----
          8 CITY-PAIR(S) IN THE LIST
CP.>
r aaa ddd
 AAA DDD           1.      1000.      1200.          6. 1 2
   2   7    700. 14   1400.

    1.   .6   0.   0.   0.   0.
CP.>
change aaa ddd ap 3 200
 AAA DDD           1.      1000.      1200.          6. 1 2
   3   3    200.   7    700. 14  1400.

    1.   .6   0.   0.   0.   0.
CP.>
.
1.>
b hd
COMMAND B HD NOT RECOGNIZED: HD ?
1.>
save
$FILE RT SAVED WITH        2 ROUTE(S)
FILE AP SAVED WITH         7 AIRPORT(S)
FILE CP SAVED WITH         8 CITY-PAIR(S)
FILE ST SAVED
1.>
end
HAVE YOU SAVED YOUR FILES ? (ENTER: Y OR SAVE).>
y
SESSION TERMINATED AT 14:51:10
```

Sample session with IFA/1   128

id	File	Page	Heading References
pt2	THESE	28	Part two: The interactive model
ipr	THESE	29	Interactive programs - Rules
fap	THESE	36	The fleet assignment procedure in FA-4
it	THESE	43	The interactive system
FGS	THESE	54	The future global system.

id	File	Page	Figure References		
fig11	THESE	11	1:	10	
struct1		THESE	38	2:	37
struct2		THESE	41	3:	39
fig23	THESE	44	4:	43	
fig24	THESE	46	5:	45, 45	
fig25	THESE	48	6:	47	
fig26	THESE	50	7:	49	
fig27	THESE	51	8:	47, 49	
fig28	THESE	56	9:	55	
fig29	THESE	59	10:	58	
fig30	THESE	61	11:	60	

id	File	Page	Footnote References	
fn11	THESE	10	1:	10
fn12	THESE	14	2:	14
fn13	THESE	16	3:	16
fn14	THESE	16	4:	16
fn15	THESE	16	5:	16
fn16	THESE	17	6:	18
fnn1	THESE	19	7:	19
fnn2	THESE	19	8:	19
fn17	THESE	20	9:	21
fn18	THESE	22	10:	21
fn19	THESE	22	11:	21

```
ffff    THESE    25    12:    25
fn21    THESE    29    13:    29
fn22    THESE    29    14:    29
ad1     THESE    30    15:    30
ad2     THESE    30    16:    30
fn23    THESE    31    17:    31
ad3     THESE    31    18:    32
fn24    THESE    40    19:    40
fn25    THESE    47    20:    47
fn26    THESE    49    21:    49
fn27    THESE    57    22:    57
fn28    THESE    60    23:    60
fn31    THESE    66    24:    66
fn32    THESE    67    25:    67
fn33    THESE    68    26:    68
new4    THESE    71    27:    71
new6    THESE    75    28:    75
new1    THESE    83    29:    83
new2    THESE    85    30:    85
```

Imbed Trace

Page 0   SYMDEF