NASA/TM–2000-209586

The Dynamic Planner: The Sequencer, Scheduler, and Runway Allocator for Air Traffic Control Automation

Gregory L. Wong

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at (301) 621-0134

- Telephone the NASA Access Help Desk at (301) 621-0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

The Dynamic Planner: The Sequencer, Scheduler, and Runway Allocator for Air Traffic Control Automation

Gregory L. Wong
Ames Research Center
Moffett Field, California

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650

Table of Contents

Acronyms

AMDT	Amount of Delay Time
AR	Acceptance Rate
ARTCC	Air Route Traffic Control Center (also known as Center)
ATC	Air Traffic Control
CM	Communications Manager
CTAS	Center-TRACON Automation System
DP	Dynamic Planner
E/DA	En Route / Descent Advisor
ETA	Estimated Time of Arrival
FAA	Federal Aviation Administration
FAF	Final Approach Fix
FAST	Final Approach Spacing Tool
FCFS	First-Come-First-Served
GUI	Graphical User Interface
IFR	Instrument Flight Rules
mf	Meter Fix
MFT	Meter Fix Crossing Time
OMA	Outer Meter Arc
OOA	Object-Oriented Analysis
OOD	Object-Oriented Design
OOP	Object-Oriented Programming
PTA	Proposed Time of Arrival
PGUI	Planview Graphical User Interface
PVD	Planview Display
RA	Route Analyzer
rwy	Runway
SCTR	Sequence Constraint Transitivity Rule
SO	Schedulable Object
STA	Scheduled Time of Arrival
TGUI	Timeline Graphical User Interface
TMA	Traffic Management Advisor
TMC	Traffic Management Coordinator
TRACON	Terminal Radar Approach Control
TS	Trajectory Synthesizer
VFR	Visual Flight Rules

The Dynamic Planner:
The Sequencer, Scheduler, and Runway Allocator for Air Traffic Control Automation

Gregory L. Wong

Ames Research Center

Abstract

The Dynamic Planner (DP) has been designed, implemented, and integrated into the Center-TRACON Automation System (CTAS) to assist Traffic Management Coordinators (TMCs), in real time, with the task of planning and scheduling arrival traffic approximately 35 to 200 nautical miles from the destination airport. The TMC may input to the DP a series of current and future scheduling constraints that reflect the operational and environmental conditions of the airspace. Under these constraints, the DP uses flight plans, track updates, and Estimated Time of Arrival (ETA) predictions to calculate optimal runway assignments and arrival schedules that help ensure an orderly, efficient, and conflict-free flow of traffic into the terminal area. These runway assignments and schedules can be shown directly to controllers or they can be used by other CTAS tools to generate advisories to the controllers. Additionally, the TMC and controllers may override the decisions made by the DP for tactical considerations. The DP will adapt its computations to accommodate these manual inputs.

This paper begins with an overview of CTAS in general and the Traffic Management Advisor (TMA) tool in particular. It lists the DP's inputs and outputs as well as the other CTAS software components required to run the DP. The heart of the paper describes the principal algorithms and data structures of the DP. This description includes a detailed discussion of the design and operation of the scheduling, sequencing, runway allocation, and miles-in-trail advisor modules of the DP. The paper concludes with a description of the methodology used in the design of the DP, the DP's current daily operational use to schedule real air traffic, and future DP research.

I. INTRODUCTION

A. Overview

The Center-TRACON Automation System (CTAS) is a set of automation tools that assist air traffic controllers and traffic management coordinators (TMCs) in the Air Route Traffic Control Center (ARTCC, also known as Center) and the Terminal Radar Approach Control (TRACON) with the planning and control of arrival air traffic [1][2][3]. CTAS is designed to generate advisories that assist air traffic controllers to improve airport capacity and reduce delays. CTAS has shown benefits such as reducing controller workload and increasing situational awareness without decreasing safety.

Currently, there are three primary tools, in various stages of development, that make up CTAS. The Traffic Management Advisor (TMA) assists Center controllers and TMCs with the scheduling of arrival aircraft within the Center's airspace [4][5][6]. The Center's airspace extends from approximately 35 to 200 nautical miles from the airport (see figure 1). The En Route/Descent Advisor (E/DA) works in conjunction with the TMA and assists Center controllers in meeting the schedules set by the TMA [7]. The Final Approach Spacing Tool (FAST) provides advisories to TRACON air traffic controllers to balance runways and sequence arrival aircraft within 40 nautical miles of and less than 10,000 feet altitude above the airport (see figure 1) [8][9][10]. CTAS also includes a capability referred to as Conflict Probe which assures that CTAS advisories are conflict free [11][12]. The Dynamic Planner (DP) is the main computational engine of the TMA and computes the sequences and schedules of arrival aircraft in the Center.

Before the 1970s, the United States air traffic control system concerned itself primarily with maintaining proper separation between aircraft. The TMCs dealt mainly with special situations such as weather and equipment failures. Under normal situations, the traffic was allowed to run freely until it became necessary to

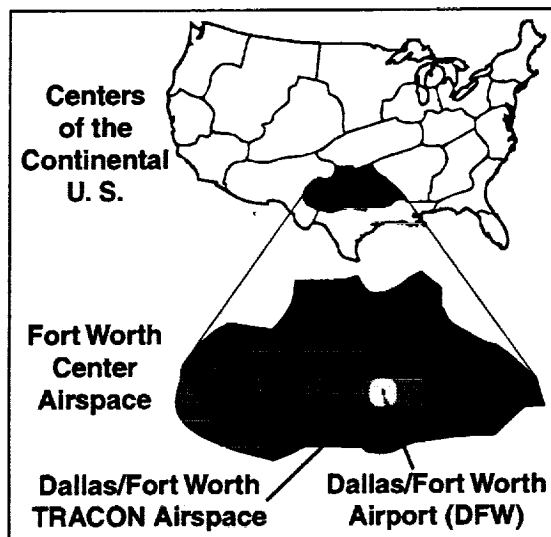hold the traffic in the terminal area. However, holding traffic at low altitudes is not fuel efficient.

Figure 1. Air Traffic Control Airspaces

The energy crisis of the 1970s resulted in an increasing emphasis on fuel efficiency. This, coupled with the air traffic controller strike of 1981, made it clear that managing the traffic flow was necessary to increase fuel efficiency, and to smooth out the workload on controllers while maintaining safety [13]. Thus, TMCs were given the added responsibility for traffic flow management.

The TMA assists the Center TMCs and controllers in a number of ways. First, the TMA helps TMCs anticipate the future traffic flow by predicting where an aircraft will be at any point in the future. The TMA also uses these predictions to compute the desired sequences and arrival times to various reference points to improve the overall flow of traffic. To help optimize the arrival times, the TMA computes runway assignments for each aircraft. In addition, the TMA assists with traffic analysis by generating statistics and reports about the traffic flow. The TMA is to be deployed nationally as part of the Federal Aviation Administration's (FAA) Free Flight Phase I program.

The DP's role in the TMA is to compute the orderly sequence, arrival times, and runway assignments to ensure a smooth flow of traffic into the terminal area. Normally, the DP sequences the aircraft so that they arrive in a first-come-first-served (FCFS) order, but TMCs can override this order by inputting sequence constraints. In addition, the TMC may input scheduling constraints which restrict the traffic flow or affect the required separation between aircraft. These scheduling constraints may be necessary due to current runway capacity, traffic density, aircraft type distribution, airport

configuration, and weather conditions. This paper describes the details of the DP's modules and algorithms.

There has been considerable effort in the development of schedulers for air traffic. Most of these include some variation of FCFS order [13][14]. There have also been attempts at designing a scheduler that specifically optimizes a cost function [15]. Some of these schedulers lack sufficient accuracy, while others are incompatible with the number of constraints that are required for use by TMCs in busy air traffic control facilities such as Ft. Worth Center. The DP is conceptually similar to many of these schedulers previously developed, but it has been designed to:

1. explicitly account for a variety of specified traffic constraints,

2. provide schedule updates that respond to the CTAS trajectory predictions,

3. optimally assign delay between the TRACON and Center, and

4. allow for the inclusion of advanced optimization features at a later date.

B. Paper Organization

This paper describes the details of the DP modules and algorithms. This paper begins with a view of the DP from outside of the DP. The inputs to the DP and the outputs from the DP are summarized. This is followed by descriptions of the other CTAS processes required to run concurrently with the DP. Next is a list of terms and definitions. The view switches to the internals of the DP with a list and explanation of the indicators and flags that the DP uses to identify special situations. This is followed by a description of the DP's principal modules. These are the main body, sequencing, scheduling, runway allocation, and miles-in-trail advisor modules. The important data structures and algorithms for each module are explained. Next, this paper describes the ETA Hovering mechanism used to compensate for errors in each aircraft's coordination fix time. This is followed by a discussion on the broadcast of the schedules computed by the DP. Next this paper describes the DP's object-oriented design methodology. Finally, this paper describes the DP's current status and future direction.

II. DESCRIPTION

A. General Description

The flight paths of some typical arrival aircraft are shown in figure 2. As each arrival aircraft flies through the Center's airspace, it crosses the outer meter arc. The outer meter arc is approximately 60 nm outside its
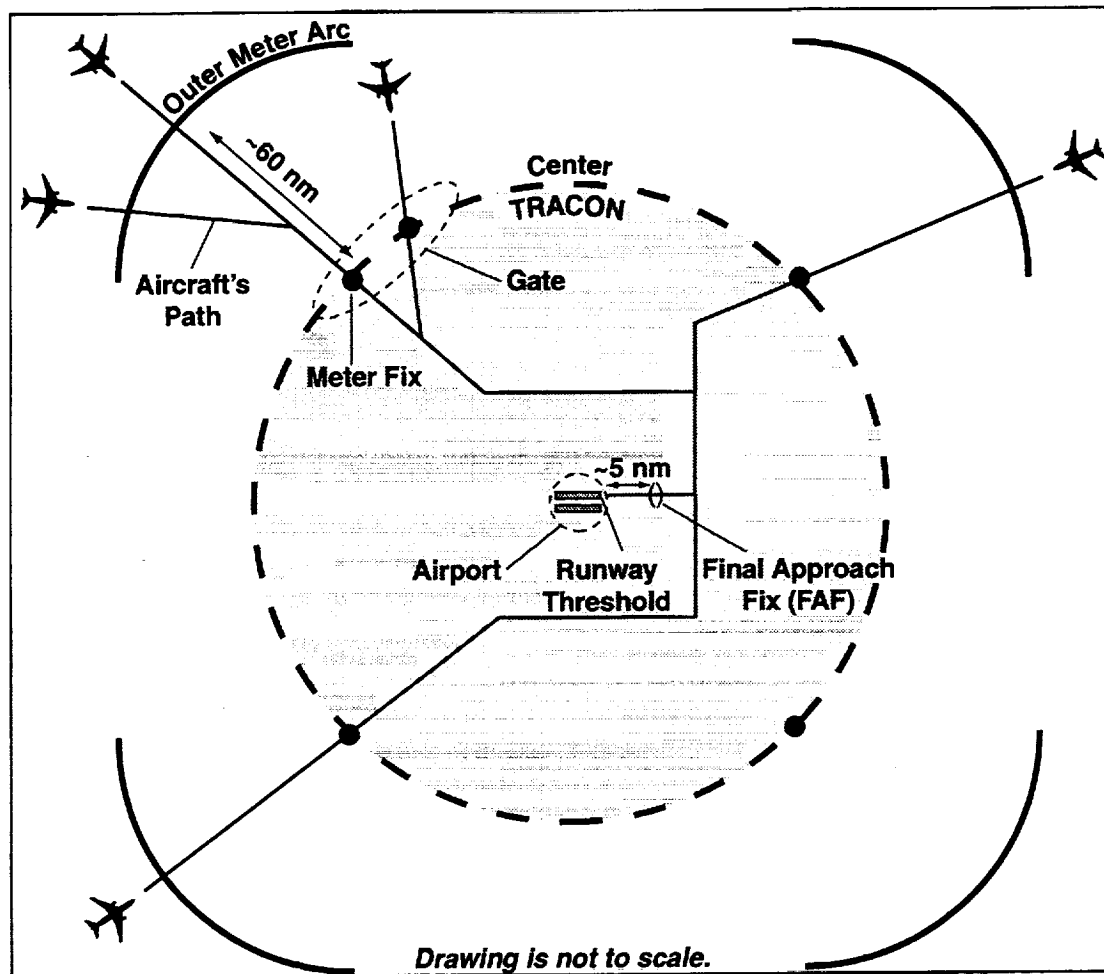
2

Figure 2. Typical Arrival Flight Paths

associated meter fix. The aircraft then crosses the meter fix which lies at the boundary between the airspaces of the Center and TRACON. For some Centers, the DP groups meter fixes together, and these meter fix groups are called gates. From the meter fix, the arrival aircraft flies through the TRACON's airspace, crosses the final approach fix (FAF), and touches down on the runway.

The DP schedules an arrival time at the outer meter arc, meter fix, FAF, and runway threshold for each aircraft. The outer meter arc[1], meter fix, FAF, and runway threshold are collectively known as Reference Points, and each of these computed arrival times is called a Scheduled Time of Arrival (STA). To account for various traffic, weather, and airport conditions, the TMC (the primary user of the DP) can control the schedule by inputting scheduling constraints such as separation

distances and acceptance rates. The DP will obey these scheduling constraints when computing the STA for each arrival aircraft. In addition, the DP sequences the aircraft to arrive at the meter fix in FCFS order. The TMC may alter this sequence by inputting specific sequence constraints. Furthermore, the DP, through a process known as Runway Allocation, will assign aircraft to runways that reduce the overall delay. The TMC may override runway assignments and manually assign an aircraft to a particular runway.

All sequencing, scheduling, and runway allocation take place while the aircraft is in the Center's airspace (approximately 25 to 300 miles from the arrival airport). Moreover, scheduling of some aircraft takes place before the aircraft have even entered Center's airspace. The DP only requires an aircraft's flight plan for scheduling. This can occur as early as 90 minutes before the aircraft enters the Center's airspace. The DP updates these sequences, schedules, and runway assignments constantly to adapt to changes in the traffic situation, changes in the environment, or in response to inputs by TMCs.

---

[1]All of the points that make up the outer meter arc are collectively treated as a single point by the TMA. Thus, an outer meter arc arrival time is the time that the aircraft will cross any point along the outer meter arc.

3

B. Inputs to the Dynamic Planner

The inputs to the DP are summarized below.

- Flight Plans
- Radar Track Updates
- Reference Point Estimated Times of Arrival (ETAs)
- Overcrossing Times
- Scheduling Constraints
- Sequence Constraints

1. Flight Plans

Flight plans contain fundamental information on each aircraft that is in, or due to enter, the Center's airspace. The flight plans are submitted by the airlines and processed by the Center's Host computer. Flight plans may also be amended by air traffic controllers while the aircraft is under their command.

The DP receives a flight plan for each aircraft being processed by CTAS. The DP uses the following flight plan information for each aircraft:

- Aircraft's identification
- Aircraft's type and characteristics
- Aircraft's planned route of flight
- Anticipated time when the aircraft will enter the Center's airspace
- Flight plan status (see below)

The following are the possible flight plan status indicators which DP receives from the Center's Host computer:

Estimated Flight Plan. The aircraft is entering the Center's airspace from an adjacent Center.

Proposed Flight Plan. The aircraft is anticipated to depart from an airport within the Center's airspace.

Departed Flight Plan. The aircraft has departed from an airport within the Center's airspace.

2. Track Updates

CTAS receives radar tracks, from the Center's Host computer, on aircraft that are in the Center's airspace. Those aircraft which have radar tracks are known as active aircraft. The DP maintains flight plans for both active and inactive aircraft.

3. Estimated Times of Arrival (ETAs)

The TMA's Route Analyzer (RA) and Trajectory Synthesizer (TS) programs work together to provide the DP with Estimated Times of Arrival (ETAs) to each of

the reference points shown in figure 2. The RA computes the horizontal route of the arriving aircraft as well as the various speed restrictions at various points along the route. The TS then takes this information, along with highly accurate aircraft and weather models, and computes a complete 4-dimensional trajectory from the aircraft's current location to touchdown at the runway threshold. From this trajectory, the RA can extract the ETAs to the reference points and send these ETAs to the DP. The ETAs are recomputed with each radar update.

From the DP's perspective, the ETAs are a prediction of when each aircraft will cross each reference point if there were no other aircraft in the airspace. Hence, the DP treats the ETAs as the earliest allowed STA. When building a schedule involving all aircraft, each aircraft's STA may have to be delayed in order to avoid a conflict with other aircraft. Thus, barring any manual interaction by the controller or TMC, each aircraft's STA will be equal to or later than the aircraft's ETA.

4. Overcrossing Times

The RA sends to the DP the time that an aircraft enters the TRACON's airspace from the Center's airspace. Within the DP, this time is referred to as the overcrossing time. Usually, this is the time that the aircraft crosses the meter fix. However, some aircraft do not actually cross a meter fix when entering the TRACON's airspace, and the RA has special logic to account for this. In either case, the DP interprets the overcrossing time as the time that the aircraft has crossed the meter fix.

5. Scheduling Constraints

When the DP computes the STAs, it obeys the scheduling constraints. These scheduling constraints are entered by the TMC to reflect the actual current and future airport capacity, mix of traffic, weather conditions, staffing level, runway topology, and air traffic control procedures. When there are no scheduling constraints, the computed STA for each aircraft will be equal to its ETA. However, when the traffic is heavy enough or the scheduling constraints are restrictive enough, the DP will begin to delay aircraft to accommodate the scheduling constraints. As a result, aircraft STAs will be later than their ETAs.

The scheduling constraints are listed below and grouped by similarity in software implementation.

- Separation Distance

  Miles-in-trail. This is the minimum horizontal distance allowed between aircraft as they cross a particular meter fix.

4

Wake Vortex Separation. This is the minimum distance allowed between aircraft as they land at a particular runway.

- Occupancy Time

   Runway Occupancy Time. This is the minimum amount of time allowed between landings at a particular runway. The TMC may enter this scheduling constraint to account for runway stopping conditions or extra time required to clear the runway.

- Acceptance Rate

   TRACON Acceptance Rate. This restricts the number of aircraft that may cross any of the meter fixes and enter the TRACON per hour.

   Meter Fix Acceptance Rate. This restricts the number of aircraft that may cross a particular meter fix per hour.

   Gate Acceptance Rate. This restricts the number of aircraft that may cross any meter fixes within a particular gate per hour. Meter fixes in the same region of airspace may be grouped together into gates. The grouping of meter fixes into gates is part of the site-adapted data of CTAS.

   Airport Acceptance Rate. This restricts the number of aircraft that may land at a particular airport per hour.

   Runway Acceptance Rate. This restricts the number of aircraft that may land at a particular runway per hour.

- Blocked Interval

   Meter Fix Blocked Interval. This is a period of time during which no aircraft are allowed to cross a particular meter fix.

   Runway Blocked Interval. This is a period of time during which no aircraft are allowed to land on a particular runway.

6. Sequence Constraints

The DP normally schedules aircraft to arrive at the meter fix in FCFS order based on their ETAs to the meter fix. However, the TMC can enter sequence constraints which force certain aircraft to be scheduled before or after other aircraft. The DP must take all such constraints into account when generating a sequence at the meter fix.

C. Outputs from the Dynamic Planner

The outputs from the DP are summarized below.

- STAs

- Runway Assignments

1. Scheduled Times of Arrival (STAs)

The DP's calculations result in a set of STAs for each aircraft at the following reference points (see figure 2).

- Outer meter arc

- Meter fix

- FAF

- Runway threshold

These STAs will meet all of the scheduling and sequence constraints. The STAs are displayed at the controllers' and TMCs' workstations to assist with traffic flow management. Additionally, DP sends the STAs to other CTAS processes that generate controller advisories.

2. Runway Assignments

In a process known as Runway Allocation, the DP will assign each aircraft to a runway. These runway assignments are designed to reduce the total delay of all aircraft and thus optimize the schedule. Additionally, the TMC can override the DP's computed runway assignment and manually assign an aircraft to a particular runway.

D. Other CTAS Components Required

The DP is a significant software component of the TMA, computing the schedule, sequences, and runway assignments based on the current air traffic situation. The DP cannot work alone. It relies on the TMA's other software components (see figure 3). The most important of these are described below.

1. Weather Data Acquisition Daemon (WDAD) and Weather Data Processing Daemon (WDPD)

The Weather Data Acquisition Daemon (WDAD) and Weather Data Processing Daemon (WDPD) acquire and process the atmospheric data that CTAS receives from an external host computer. Currently, atmospheric data is provided by the National Centers for Environmental Prediction (NCEP).
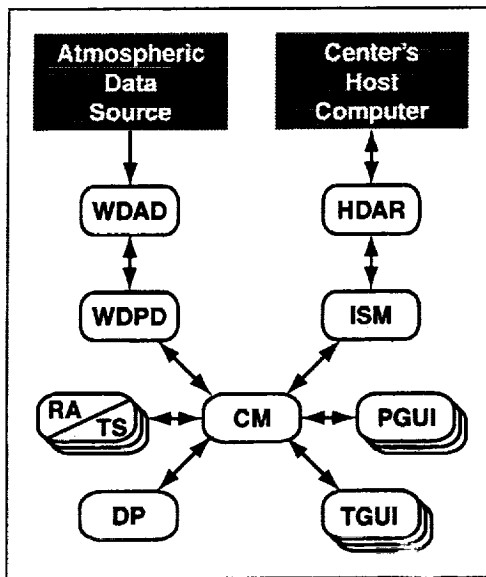
Figure 3. TMA Software Compo? .ts

2. Host Data Acquisition and Routing (HDAR) and Input Source Manager (ISM)

The Host Data Acquisition and Routing (HD \R) process and the Input Source Manager (ISM) serve as the interface between the Center's Host computer and CTAS. This is a two-way interface through which flight plan information and radar tracks are sent from the Host to CTAS while schedules and flight plan amendments are sent from CTAS to the Host.

3. Communications Manager (CM)

The Communications Manager (CM) controls the communication between each CTAS software process and maintains a central database to keep all the processes in synch with one another. All of the DP's input and output data (such as schedules and flight plan amendments) are received from, or sent to, the CM. It is the CM's responsibility to distribute the data to the proper CTAS processes. In addition, CM maintains a centralized database that includes all of the scheduling constraints entered by the user. Every time a CTAS process is started and connects to the CM, the database is transferred to the newly connecting process to synchronize it with the rest of CTAS.

4. Trajectory Synthesizer (TS)

The Trajectory Synthesizer (TS) is the computational engine of CTAS [16][17]. It provides the other CTAS processes with accurate 4-dimensional trajectories and the associated ETAs at various reference points for each aircraft. Inputs to the TS include aircraft model data for all aircraft types (including aerodynamic and propulsion characteristics, and preferred speeds), airspace structure data, and near real-time weather information. In addition, the Route Analyzer (RA) (see section II.D.5) provides the TS with each aircraft's initial condition, horizontal route of flight, desired end conditions, and intermediate speed and altitude constraints. The output of the TS consists of nominal, slow, fast, and meet-time ETAs to various reference points, aircraft ground speeds at the reference points, and the nominal 4D trajectory for each aircraft. Of particular importance to the DP's calculations are the ETAs to, and the ground speeds at, each of the following reference points.

- Outer Meter Arc
- Meter Fix
- Final Approach Fix
- Runway Threshold

The thoroughness and accuracy of the TS is a significant improvement in generating ETAs over that of previous engineering efforts. The DP is a beneficiary of this improvement and can create realizable schedules given the highly accurate ETAs provided by the TS.

5. Route Analyzer (RA)

The Route Analyzer (RA) generates all possible realistic horizontal routes for an aircraft. These routes extend from the aircraft's current position to its end point based on a set of site-adaptable analysis rules. These horizontal routes are computed for each aircraft every time a radar update is received. The horizontal routes, along with the aircraft's initial condition, desired end conditions, and intermediate altitude and speed restrictions, determined by the RA, are passed along to the TS. From the resulting TS output, the RA extracts the fast, nominal, and slow ETAs to every eligible runway threshold, FAF, assigned meter fix and outer meter arc. Currently, the DP only uses the nominal ETAs. The fast and slow ETAs are reserved for future scheduling research. Additionally, the RA extracts the ground speeds at these points. The RA then sends these ETAs and ground speeds to the DP. Finally, when the RA detects that an aircraft has crossed its meter fix or has otherwise entered the TRACON's airspace from the Center's airspace, it sends the DP the overcrossing time of that aircraft.

6. Timeline and Planview Graphical User Interfaces (TGUI and PGUI)

The Timeline Graphical User Interface (TGUI) and the Planview Graphical User Interface (PGUI) provide the principal means by which the user may interact with the DP. All scheduling and sequencing constraints and any other manual inputs to the DP are done through the

various panels and displays provided by the GUIs. The DP's output, such as schedules and aircraft status, is displayed to the user on the GUIs. The GUIs also provide other information which serve to enhance the TMA user's situational awareness.

E. Terminology

The following terms and concepts are used to describe the DP. Some terms may be specific to the DP and are not part of the air traffic control (ATC) vernacular.

1. Blocked Slot

Blocked slots can best be thought of as phantom aircraft. A blocked slot may be used to hold open a space where a real but untracked aircraft is anticipated to be. The TMC may create a blocked slot relative to a meter fix or a runway. For a meter fix blocked slot, the user specifies the blocked slot type, assigned meter fix, arrival airport, and ETA to the meter fix. For a runway blocked slot, the TMC specifies the blocked slot type, assigned runway, and ETA to the runway. The blocked slot type is selected by the TMC from a set of engine type and weight class combinations. A flight plan is created by the CM using an aircraft model representative of the engine type and weight class the user has selected (see table 1). This flight plan is then distributed throughout CTAS as an inactive aircraft (since there are no tracks for it). For a meter fix blocked slot, the TMC specifies the meter fix ETA of the blocked slot. The ground speeds at the meter fix, FAF, and runway threshold, and the ETAs to the FAF and runway threshold are computed by the RA based on nominal trajectory information for the representative aircraft model. For a runway blocked slot, the TMC specifies the runway ETA. The ground speed at the runway is computed by the RA based on nominal trajectory information for the representative aircraft model.

2. Schedulable Objects (SOs)

The DP computes STAs for Schedulable Objects (SOs). Aircraft and blocked slots are collectively known as SOs.

3. Acceptance Rate Interval

Although the acceptance rate is expressed as the number of SOs per hour, the actual time period used by the DP is a fraction of an hour. This time period is known as the Acceptance Rate Interval and varies from site to site. For example, at Denver airport (DEN), the Acceptance Rate Interval is 15 minutes. In contrast, the Acceptance Rate Interval at Dallas-Ft. Worth airport (DFW) is 10 minutes. Thus, an acceptance rate of 96 SOs per hour is treated as 24 SOs per 15 minute period at DEN and 16 SOs per 10 minute period at DFW. The acceptance rate must be

satisfied over any window the size of the Acceptance Rate Interval. For example, at DFW, any 10 minute window should not contain more than 16 SOs.

Table 1. Blocked Slot Aircraft Types

Blocked Slot Type	Representative Aircraft Model	FAA Designation
small piston	Cessna 172	C172
small turboprop	Beechcraft King Air 200	BE20
large turboprop	Embraer EMB-120	E120
large jet	Boeing 727	B727
heavy jet	Boeing DC-10	DC10
Boeing 757	Boeing 757	B757

4. Stream Classes

Center arrival traffic is categorized into stream classes. Each stream class contains SOs with similar scheduling characteristics. Currently, this categorization is based on engine type, destination airport, and assigned meter fix.

5. Super Stream Classes

During runtime, stream classes with similar scheduling characteristics may be grouped into super stream classes. Every stream class will be included in one super stream class though several stream classes may be placed in the same super stream class.

6. Reference Point

The runway threshold, FAF, meter fix, and outer meter arc are all reference points. For each SO, the DP computes STAs to various reference points, and the scheduling constraints are applied at these reference points.

7. Transition Time

The transition time is the time an SO takes to fly from one reference point to another if there were no other SOs in the system. The transition time is computed by taking the ETA of the destination reference point and subtracting the ETA of the source reference point. For example, if the meter fix ETA is 1045Z and the FAF ETA is 1102Z for a particular SO, the transition time from the meter fix to the FAF is 17 minutes.

8. Flight Rules

The DP's computation of STAs to the runway depends on the flight rules in effect for the airport's configuration. The two flight rules handled by the DP are Instrument Flight Rules (IFR) and Visual Flight Rules (VFR). These flight rules, within the DP's context, are defined below.

9. Instrument Flight Rules (IFR)

Under IFR conditions, the controllers guide the aircraft all the way to touchdown. Hence, the DP will compute the STA for each aircraft at the runway threshold to meet the runway and airport scheduling constraints. The FAF STA is computed by subtracting the transition time between the FAF and the threshold from the threshold STA.

10. Visual Flight Rules (VFR)

Under VFR conditions, the controllers guide the aircraft to the FAF. H·ice, the DP will compute the STA for each aircraft a ·he FAF to meet the runway and airport scheduling constraints. The threshold STA is computed by adding the transition time between FAF and the threshold to the FAF STA.

11. Configurations

The TMC specifies an airport's current and future configurations and the time of the future configuration changes. The configuration defines which runways are active, which runways are dependent on each other, and which flight rules are in effect. When scheduling to dependent runways, the DP treats the runways within a dependent set as if they were a single runway.

Associated with each configuration is a flow parameter set. One flow parameter set is selected for a configuration from a group of flow parameter sets available for that configuration.

12. Flow Parameter Sets

Flow parameter sets are scheduling constraint macros associated with each airport configuration. A flow parameter set is designed to set a number of scheduling constraints to control the traffic flow rate to the airport.

13. STA Freeze

An SO will become STA-Frozen when its ETA at the meter fix is less than or equal to M minutes in the future. The value of M is known as the STA Freeze Horizon and varies from stream class to stream class and from site to site. A typical value is 19 minutes for jets. Inside the STA Freeze Horizon, an SO's STA will not change when the schedule is updated. Exception: STA-Frozen SOs will have their STAs recomputed and, possibly, changed

in response to scheduling events that correspond to Scheduling Modes 1 through 5 (see table 2).

14. Sequence Freeze

An SO will become Sequence-Frozen when its ETA at the meter fix is less than or equal to N+M minutes in the future. The value of N is known as the Sequence Freeze Horizon and M is the STA Freeze Horizon. Both N and M vary from stream class to stream class and from site to site. A typical value for N is 5 minutes. Because both N and M are non-negative numbers, aircraft that are STA-Frozen are also Sequence-Frozen.

Unlike STA-Frozen SOs, Sequence-Frozen SOs can have their STAs changed during the scheduling process. For Scheduling Modes 4, 5, 8, and 9 (see table 2), Sequence-Frozen aircraft will be scheduled such that they maintain their super stream class sequence. For example, if aircraft A and B are both Non-Sequence-Frozen aircraft in the same super stream class, and A's ETA is earlier than B's ETA, then A will be placed in the sequence ahead of B. Later, if A and B become Sequence-Frozen and A's ETA becomes later than B's ETA, then A will still be placed in the sequence ahead of B.

Sequence Freeze only applies to meter fix sequences. Sequences at the runway threshold or FAF may change. Also, some events will result in the resequence of Sequence-Frozen SOs. These events correspond to Scheduling Modes 1, 2, 3, 6, and 7 (see table 2).

15. Scheduling Modes

Each scheduling event is mapped to a scheduling mode. The scheduling mode influences which SOs are rescheduled. Some events affect all SOs, while some events affect only those SOs whose ETA is at, or later than, a certain point in time. Additionally, some scheduling events require that all or some of the SOs be resequenced as part of the scheduling process. For each possible freeze state (STA-Frozen, Sequence-Frozen, and Other), the scheduling mode indicates whether or not sequences from the previously computed schedule are to be maintained in the new schedule. The scheduling modes are listed in table 2.

F. Special SO Indicators

There are a number of indicators or flags associated with each SO. These flags may affect how the SO is treated during scheduling and runway allocation. Some of these flags may be set as a result of the user's interaction with the DP. Additionally, some flags are set by the DP automatically in response to changes in data or the receipt of various events.

Table 2. Scheduling Modes

Scheduling Mode #	Which SOs Included[a]	STA-Frozen[b]	Sequence-Frozen	Other[c]
1	single SO	disregard sequence[d]	disregard sequence[e]	disregard sequence
2	all SOs	disregard sequence	disregard sequence	disregard sequence
3	all SOs from[f]	disregard sequence	disregard sequence	disregard sequence
4	all SOs	maintain sequence	maintain sequence	disregard sequence
5	all SOs from	maintain sequence	maintain sequence	disregard sequence
6	all SOs	no change in STA	disregard sequence	disregard sequence
7	all SOs from	no change in STA	disregard sequence	disregard sequence
8	all SOs	no change in STA	maintain sequence	disregard sequence
9	all SOs from	no change in STA	maintain sequence	disregard sequence
10	all SOs	no change in STA	no change in STA	disregard sequence
11	all SOs from	no change in STA	no change in STA	disregard sequence

a."Which SOs Included" refers to the SOs that are selected for inclusion in the next scheduling update.

b.STA-Frozen SOs that are included in a scheduling mode update under a particular scheduling mode can have their STA changed by that scheduling update.

c."Other" refers to SOs that are neither STA-Frozen nor Sequence-Frozen.

d.If the single SO is STA-Frozen, it will be rescheduled disregarding its sequence, but the other STA-Frozen SOs will be left alone.

e.The phrase "disregard sequence" means that the sequence from the last scheduling state is not a sequence constraint for the next scheduling update. All other sequence constraints are still observed.

f.Not all SOs are rescheduled. An SO is rescheduled if its STA or ETA is equal to or later than a point in time as dictated by the scheduling event.

1. Departed Aircraft

Arrival aircraft that are to depart from airports within the Center's airspace are represented in the system by proposed flight plans. Proposed flight plans create uncertainty because their filed departure times are often inaccurate. Aircraft have been known to actually depart up to three hours earlier or later than the time indicated by the flight plan. These highly inaccurate departure times result in highly inaccurate ETAs. If the DP were to schedule the aircraft using such a faulty ETA, the resulting STA may be unrealizable. Therefore, these aircraft require special treatment.

The DP will not schedule proposed flight plans unless the TMC has manually input the aircraft's departure time. When the TMC is informed of the accurate departure time of the aircraft, the TMC enters the departure time into the TGUI. The TGUI informs the DP, and the DP will set a status flag for that aircraft indicating that the aircraft has been "manually departed." Additionally, the TGUI will pass along the departure time as a flight plan amendment to the aircraft's coordination time. The DP will dispose of any previously computed ETA for the aircraft and will wait to receive a new ETA based on this coordination time. When the new ETA is received, DP will be able to compute an STA for the aircraft and thus hold a slot for that aircraft until the aircraft becomes airborne and radar tracks are received. Once track data is received, the aircraft is treated the same as any other tracked aircraft.

2. Expired Aircraft

Normally, after an aircraft lands or is otherwise removed from the Center Host computer's database, the Host computer instructs CTAS to delete the aircraft's flight plan. Occasionally, CTAS does not receive such a message, and the flight plan remains in CTAS. If this

9

aircraft were left unchecked, the DP would continue to schedule the aircraft as though it were simply an inactive flight plan, and this aircraft would take up a slot in the schedule. To assure that such an aircraft will not take up a slot in the schedule, the DP will flag the aircraft as expired and exclude it from the scheduling and runway allocation processes. An aircraft is flagged as expired if all of the following criteria are true:

- Aircraft is inactive.

- Aircraft is not a proposed flight plan.

- The aircraft's runway ETA is in the past.

Only aircraft can be flagged as expired. It is not necessary for a blocked slot to expire because the CM will delete a blocked slot's flight plan after a certain amount of time.

3. Pop-up Aircraft

An aircraft that is flagged as a pop-up is excluded from the scheduling and runway allocation processes. Note, only aircraft can be flagged as a pop-up. Blocked slots are never flagged as pop-ups.

The DP's initial design and implementation included provisions for pop-up aircraft.[2] However, the current DP implementation does not flag any aircraft as a pop-up. The pop-up functionality may be reactivated as part of a future enhancement.

4. Priority Aircraft

The user may designate an aircraft as being a priority aircraft. This functionality is used in emergency situations where the aircraft must land as soon as possible. A priority aircraft will have a different ETA computed for it by the RA. This ETA is based on a quicker and more direct route to the runway threshold. Priority aircraft will normally be scheduled at their ETA unless this puts them in conflict with SOs not being rescheduled, manually scheduled SOs, or other priority aircraft. The position of a priority aircraft in the sequence is irrelevant when scheduling to the ETA.

5. Landed Aircraft

When an aircraft's track is within a certain distance and altitude of its arrival airport, the CM informs all CTAS processes that the aircraft has landed. This is necessary because the Center's Host computer does not inform CTAS when an aircraft has landed.

---

[2]In the DP's original design an aircraft was defined as a pop-up if the first ETA associated with the aircraft's first radar track occurred within the STA Freeze Horizon.

Landed aircraft are not eligible for scheduling nor runway allocation. However, the number of aircraft that have landed in the past hour must be counted when applying an acceptance rate scheduling constraint.

6. Manually Scheduled SOs

The user may manually set the STA of an SO. The user may supply the STA to a reference point such as the meter fix or runway threshold, and the DP will compute the STA to the other reference points. The difference between the ETAs to the various reference points is added or subtracted from the user-entered STA to derive the STA to the other reference point. The DP will automatically freeze the STA of a manually scheduled SO. If the manually set STA places the SO in conflict with other STA-Frozen SOs, then the conflict will not be resolved. On the other hand, Non-STA-Frozen SOs will be scheduled in such a way as to avoid a conflict with the manually scheduled SO. The DP's philosophy regarding manually scheduled SOs is that the user has assigned the STA to an SO, and it is up to the DP to schedule the remaining SOs around the manually scheduled ones.

7. Suspended SOs

The user may suspend or unsuspend an SO. A suspended SO is not scheduled, and its slot is given up when the SO is first suspended.

8. Holding Aircraft

Aircraft which have been placed in holding require special handling by the scheduler. Some holding aircraft are instructed, by the controller, to follow a holding pattern over a meter fix until it is taken out of holding. As a result, the aircraft's track alternates between the Center's airspace and the TRACON's airspace. The DP compensates for this by treating aircraft that are in holding as if they are in the Center's airspace even if the aircraft's track indicates that the aircraft is in the TRACON's airspace. However, currently, no CTAS process informs the DP that an aircraft is in holding. The detection of an aircraft in holding is an area of future research.

G. The Main Body

All of the DP's activity is event driven, and the DP's main body controls the program flow based on these events. Most events come from other CTAS processes and are received via messages while other events are generated by the DP itself.

When a message is received from another CTAS process, the DP will update its databases as appropriate and then execute one of the following three activities.

Do nothing. This occurs when the message simply requires that the DP update its database.

Place a rescheduling or runway allocation event on the list of pending events for processing later. Some events of the same event type occur in bunches. To improve computing performance, these events are collected in a pending list and are processed as a group only after an event of a contrasting event type is received.

If the pending list is full[3] or if the current event differs from those in the pending list, then the appropriate rescheduling or allocation actions on one or more aircraft for the pending events are executed. The pending list is then flushed, and the current event is placed in the pending list.

Carry out an immediate reschedule or runway allocation. If there are events in the pending list, then rescheduling or runway allocation is executed on one or more aircraft for the pending events. The pending list is then flushed, and rescheduling or runway allocation is carried out on one or more aircraft for the current event.

The DP will send messages to the other CTAS processes as a result of rescheduling or runway allocation. Runway allocation activity is always followed by rescheduling in the DP. If an aircraft's assigned runway is changed as a result of runway allocation, then a flight plan amendment indicating the runway change is sent by the DP to the other CTAS processes. Similarly, if rescheduling an aircraft shifts the aircraft's STA from a time before a configuration change to a time after the configuration change, then a flight plan amendment indicating that the aircraft is under the influence of a different airport configuration is sent by the DP to the other CTAS processes. After rescheduling is complete, a schedule message is sent to all of the other processes indicating the newly computed STAs. The schedule message will also include information about the changes in the aircraft status as a result of rescheduling.

In addition to responding to messages sent by other CTAS processes, the DP generates its own internal rescheduling or runway allocation triggering events. If no rescheduling has been executed for six seconds, then the DP will carry out a reschedule known as a Periodic Reschedule. If an aircraft is about to have its STA frozen, then a runway allocation event for that aircraft is generated internally. Rescheduling may also be triggered internally when an aircraft's ETA is hovered. The mechanism of ETA hovering is described in a later section.

---

[3]The capacity of the pending list is site-dependent. Currently, for all adapted sites, the list can hold 50 pending events.

H. Sequencing

An integral part of scheduling is sequencing. A sequence is the order in which aircraft are to arrive at a particular scheduling reference point. The DP handles sequencing through its Sequencer and Sequence Constraint modules.

Aircraft to be scheduled are first sequenced in an FCFS order within each stream class based on their ETAs to the meter fix. However, depending on the scheduling mode, the sequence may be further restricted so that aircraft which are Sequence-Frozen maintain their sequence relative to other Sequence-Frozen aircraft. Furthermore, the user may input sequence constraints to the DP which force certain aircraft to be scheduled before or after other aircraft. The sequencer must take all of these restrictions into account when generating a sequence at the meter fix.

1. Sequence Constraints

The sequence may deviate from the FCFS rule as a consequence of sequence constraints. A sequence constraint is an instruction to the DP to force certain aircraft to follow behind other aircraft. Currently, sequence constraints are entered by the users via CTAS's graphical user interfaces, PGUI and TGUI. These sequence constraints are sent to CM, and CM forwards them to the DP. So, from the DP's point of view, the sequence constraints are always generated by other CTAS processes. Sequence constraints always involve two aircraft, have a time stamp and priority associated with them, and occur in two varieties.

The first variety of sequence constraint is referred to as a "direct" constraint. If aircraft B is constrained directly behind aircraft A, then the DP will create a sequence that places aircraft B immediately following A even if this means delaying either aircraft so that B can follow behind A. A direct constraint is notated here with a double directed line pointing to the aircraft ahead. For example, the sequence constraint just described is shown in figure 4.

Figure 4. Direct Constraint

The second type of constraint is an "indirect" sequence constraint. If aircraft B is constrained indirectly behind A, then the DP will create a sequence that places B somewhere after A. This constraint allows for other aircraft to be sequenced behind A and in front of B so long as B is somewhere behind A. An indirect constraint is notated here with a single directed line pointing to the

aircraft ahead. For example, the sequence constraint just described is shown in figure 5.
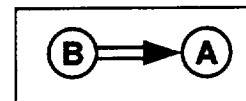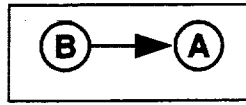

Figure 5. Indirect Constraint

Since sequences are computed independently for each super stream class, any sequence constraints involving aircraft that are in different super stream classes are ignored. For example, suppose aircraft B, a turbo prop, is constrained directly behind aircraft A, a jet. If the TMC has not combined these two engine types into the same super stream class, this sequence constraint will be ignored.

2. Sequence Constraint Deconfliction

The most complicated function related to sequencing is sequence constraint deconfliction. If the DP has been issued sequence constraints that conflict with each other, the DP applies an order of precedence to determine which constraints are active and which are inactive at the time of the sequencing. Each sequence constraint received by the DP has a relative priority associated with it. In the current NASA proof-of-concept software, sequence constraints received from the TGUI have a priority of 10 and those received from the PGUI have a priority of zero[4]. If two sequence constraints are in conflict, then the sequence constraint with a higher priority number will have precedence. If the sequence constraints have the same priority, then the sequence constraint entered more recently is used.

The deconfliction of sequence constraints is dynamic. That is, two sequence constraints might be in conflict during one scheduling cycle, but they may not be in conflict during a later scheduling cycle. Therefore, a sequence constraint is not removed from the DP unless the DP is explicitly told to remove it, or if one of the aircraft listed in the constraint has been removed from the system. Instead, sequence constraints that are in conflict with constraints that have a higher priority (or are newer if the priorities are equal) are made inactive for that scheduling cycle.

Deconflicting sequence constraints requires building a directed graph called the Sequence Constraint Graph. Each node in the graph (represented by a circle in the accompanying figures) represents an SO involved in a

---

[4]The sequence constraint priority values are contained within the sequence messages that the DP receives from other CTAS processes. These values may be changed to suit the operational environment without changing the DP.

constraint, and each SO can appear, at most, once in the graph. Each directed edge (represented by an arrow in the accompanying figures) extends from the SO behind in the sequence constraint to the SO ahead in the sequence constraint. Additionally, each edge indicates if the sequence constraint is a "direct" constraint (represented by double lines in the following diagrams) or an "indirect" constraint (represented by a single line in the following diagrams).

Each node in the graph must obey certain fan-in and fan-out limits. The allowable fan-in for a particular node (the number of directed edges pointing to a particular node) is either one direct constraint or zero or more indirect constraints. Similarly, the allowable fan-out for a particular node (the number of directed edges pointing away from a particular node) is either one direct constraint or zero or more indirect constraints. Finally, no edge may begin from and end at the same node. See figure 6 for an example of a valid Sequence Constraint Graph and figure 7 for three examples of invalid Sequence Constraint Graphs.
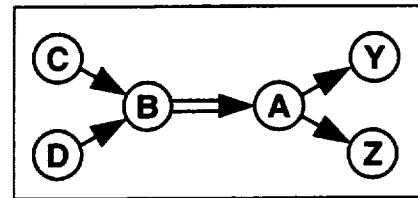
Figure 6. Valid Sequence Constraint Graph
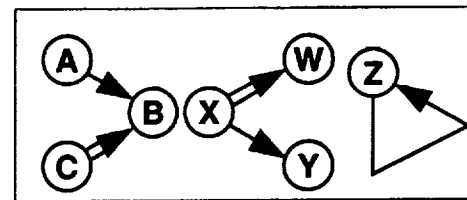
Figure 7. Examples of Invalid Fan-in and Fan-out

The process of deconflicting sequence constraints consists of iterating over each constraint q_i in order from highest to lowest priority. Within each priority, the order is from newest to oldest. Each sequence constraint q_i is compared with each of the sequence constraints q_j that has been examined during a previous iteration and has been marked as active. Each q_i is compared against each q_j using the following three checks.

1. Individual Check

2. Chain Check

3. Graph Update

If q_i fails any of these checks when compared against any of the previously examined active sequence constraints q_j, then q_i is marked as inactive. An inactive sequence constraint will not affect the sequence during the current scheduling cycle. On the other hand, if q_i passes all of the preceding checks against each q_j, then q_i is marked as active. An active sequence constraint will influence the sequence during the current scheduling cycle, and the remaining sequence constraints to be checked will be examined for conflict with each of the active constraints. Each of these checks is described in greater detail below.

Individual Check. Several simple quick checks are conducted between the sequence constraint q_i and the sequence constraints that have already been accepted during the preceding iterations. These checks are broken down into two phases. During the first phase of individual checks, the constraint q_i is compared on a one-to-one basis with each constraint q_j which has already been accepted. During the second phase of individual checks, the constraint q_i, if it is an indirect sequence constraint, is compared against the indirect constraints contained within the Sequence Constraint Graph. The Sequence Constraint Graph is the set of sequence constraints q_j which have been deconflicted and are currently active. If the comparisons show that q_i is redundant or contradicts a previously accepted constraint q_j, then q_i is made inactive. The specific cases which result in making q_i inactive are as follows.

If a previously accepted constraint q_j is exactly like the constraint q_i, then q_i is made inactive. This is known as rejection by redundancy. For example, if q_j and q_i are those shown in figure 8, then q_i is made inactive in favor of the higher priority or newer constraint which was previously accepted.
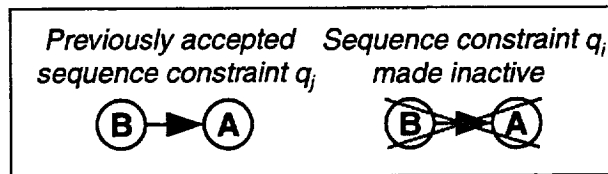

Figure 8. Redundant Constraint

If a previously accepted constraint q_j has the same two SOs as q_i but in reverse order, then q_i is made inactive. For example, figure 9 shows q_i which is the reverse of a previously accepted constraint q_j. Due to this contradiction, the q_i is made inactive.
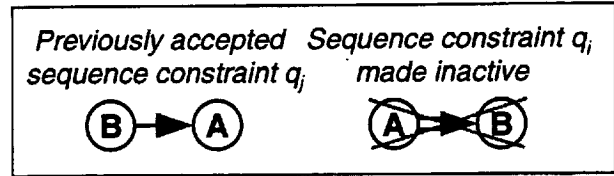

Figure 9. Reverse Constraint

If the combination of the constraint q_i and a previously accepted constraint q_j would result in one SO being constrained directly ahead of or behind two different SOs, then q_i is made inactive. For example, figure 10 shows a case where the combination of q_i and a previously accepted constraint q_j constrains two different SOs (B and C) directly behind the same SO (A). Also, figure 11 shows a case where the combination of q_i and q_j constrains the same SO directly behind two different SOs. In both cases, q_i is made inactive.
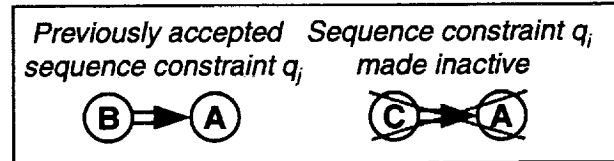
Figure 10. Two SOs Constrained Behind One SO
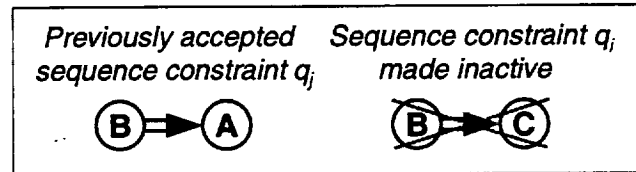
Figure 11. SO Constrained Directly Behind Two SOs

If a previously accepted constraint q_j constrains one SO directly ahead of or behind a second SO while the constraint q_i only constrains the first SO indirectly ahead of or behind the second SO, then q_i is made inactive (rejection due to redundancy). In figure 12, the constraint q_i does not add any more information because the previously accepted constraint is more restrictive. Therefore, q_i is made inactive.
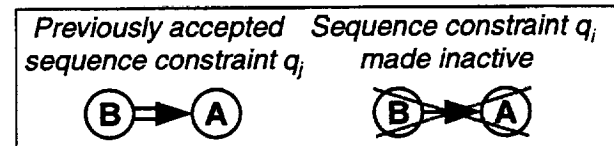
Figure 12. Less Restrictive Constraint

The second phase of the individual checks consists of comparing the indirect constraint q_i against the previously accepted indirect constraints that are

contained in the current Sequence Constraint Graph. The Sequence Constraint Graph is created and updated during the Graph Update step of sequence constraint deconfliction. Note that this second phase is only concerned with indirect constraints. If q_i is a direct constraint, then it passes this second phase of individual checks. Furthermore, the construction of the Sequence Constraint Graph may alter some of the indirect sequence constraints that are added to the graph. The purpose of this second phase of individual checks is to compare the indirect constraint q_i against those altered indirect constraints contained within the graph. Those constraints contained within the graph which are unaltered have already been compared against q_i in the first phase of the individual checks described above.

If the indirect constraint q_i is the same as an indirect constraint within the graph, then q_i is made inactive (rejection due to redundancy). An example of this case is shown in figure 13.
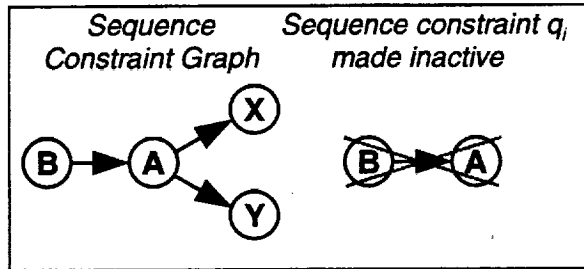

Figure 13. Redundant Constraint

If the indirect constraint q_i is the reverse of an indirect constraint within the graph, then q_i is made inactive. In the example shown in figure 14, q_i is made inactive because it is the reverse of a constraint contained within the graph.
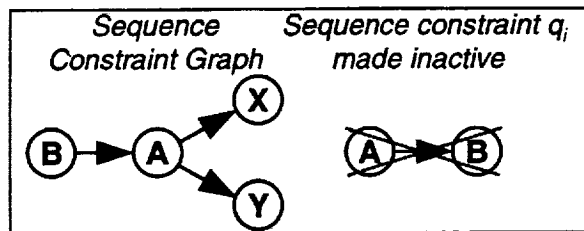
Figure 14. Reverse Constraint

Chain Check. Next, the constraint q_i is checked against the Sequence Constraint Graph to see if q_i conflicts with a chain of constraints contained within the graph. If either the preceding SO or the following SO (or both) in q_i is not contained in the Sequence Constraint Graph, then q_i passes the Chain Check.

If both the ahead SO and the behind SO in the constraint q_i are contained in the Sequence Constraint Graph, then

more detailed comparisons between q_i and the graph are conducted. For each SO contained in the Sequence Constraint Graph, a list containing all SOs constrained anywhere ahead of that SO is created. The constraint q_i is then compared against these lists to determine if q_i contradicts or duplicates the information contained in these lists. Cases where q_i is made inactive are described below.

If the constraint q_i is the reverse of a constraint chain contained in the Sequence Constraint Graph, then q_i is made inactive. An example is shown in figure 15.
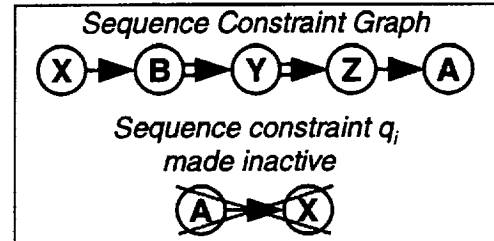
Figure 15. Reverse of a Chain Constraint

If the constraint q_i is a duplicate of a constraint chain contained in the Sequence Constraint Graph, then the current constraint **may** be redundant. For example, in figure 16, q_i is made inactive because it is redundant.
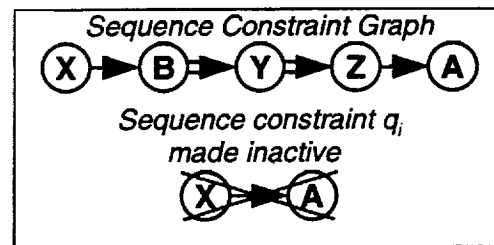
Figure 16. Redundant with a Chain Constraint

An exception to the case described above is shown in figure 17. If q_i is a direct constraint, then it replaces a less restrictive indirect constraint involving the same SOs when there are no intervening SOs.
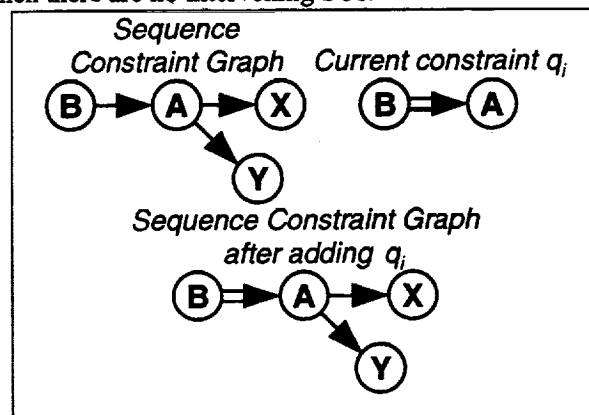
Figure 17. Adding a More Restrictive Constraint

14

Graph Update. In order to remain within the fan-in/fan-out limits, the Sequence Constraint Transitivity Rule (SCTR) is used to adjust an existing Sequence Constraint Graph when new sequence constraints are inserted. The SCTR is detailed in the sidebar at right.

Building the directed Sequence Constraint Graph consists of examining each sequence constraint q_i in order, beginning with the newest/highest priority sequence constraint and ending with the oldest/lowest priority sequence constraint. Each acceptable sequence constraint q_i is added to the graph, and adjustments are made to the graph to comply with the fan-in/fan-out limitations. Each constraint falls into one of the cases described below.

Case 1: The sequence constraint q_i is already in the graph.

The constraint q_i is ignored in favor of the newer/higher priority constraint already in the graph (see figure 18).
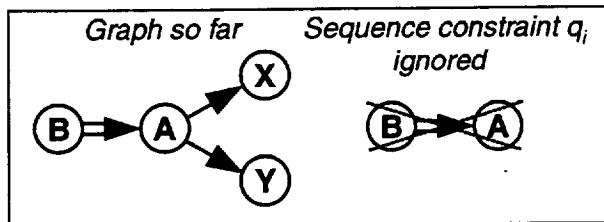
Figure 18. Trying to Add an Existing Constraint

Case 2: Neither the SO ahead nor the SO behind in constraint q_i appears in the graph.

A new node is created for the SO ahead and the SO behind in the constraint q_i, and the constraint itself is added. In this case, q_i is disjoint from the Sequence Constraint Graph thus far (see figure 19).
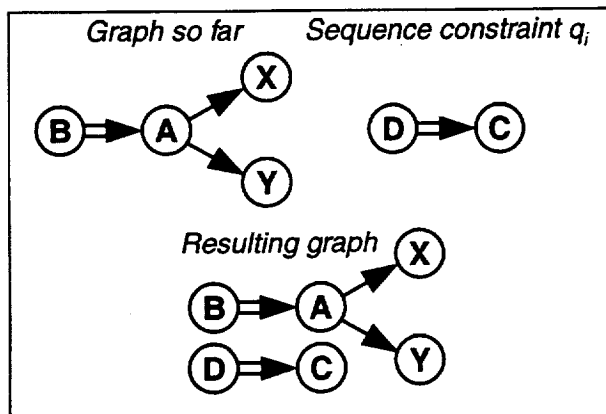
Figure 19. Disjoint Constraint Added to the Graph

The Sequence Constraint Transitivity Rule (SCTR)

Let A, B, C, D, X, Y, and Z be distinct SOs.

If B is constrained directly behind A, and C is constrained indirectly behind A, then C is constrained indirectly behind B (see figure 20).

Similarly, if Y is constrained directly ahead of X, and Z is constrained indirectly ahead of X, then Z is constrained indirectly ahead of Y (see figure 21). This may be extended to include chains of direct sequence constraints (see figure 22)
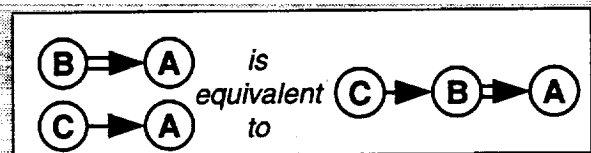
Figure 20. SCTR Example #1
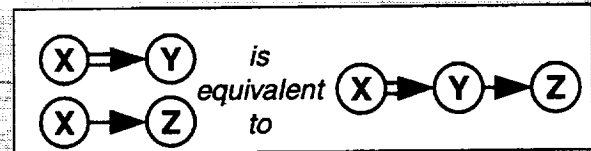
Figure 21. SCTR Example #2
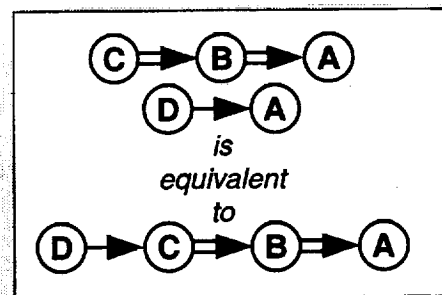
Figure 22. SCTR Direct Constraint Chain Example

Case 3: The constraint q_i is an indirect constraint, and either the SO ahead or the SO behind in the constraint (but not both) is already in the graph.

A new node is created for the SO in the constraint q_i that is not already in the graph. An edge is added that links the new node with the SO that is in q_i and already in the graph (see figure 23).
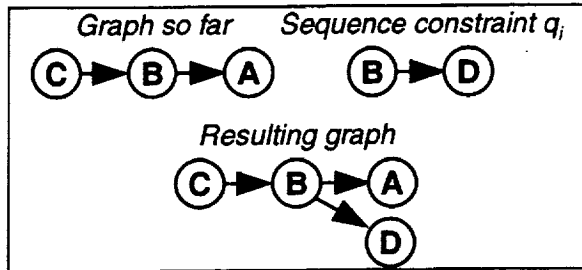
Figure 23. Indirect Constraint Added to the Graph

Simply inserting q_i could exceed the fan-in/fan-out limit because of direct constraints in the graph. In this case, the SCTR is applied, and the new edge is added at the end of the chain of direct constraints (see figure 24 and figure 25).
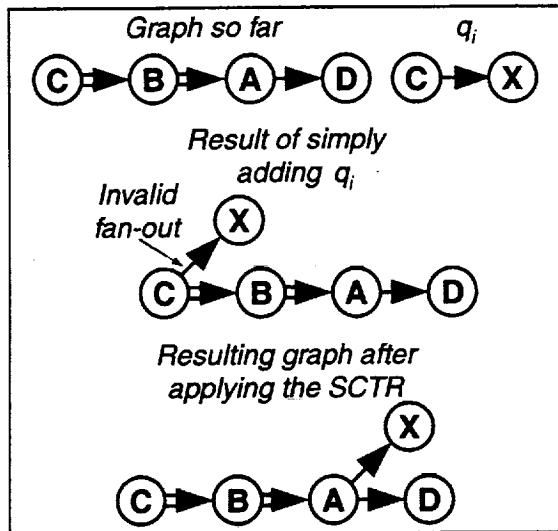
Figure 24. SCTR Applied to an Indirect Constraint
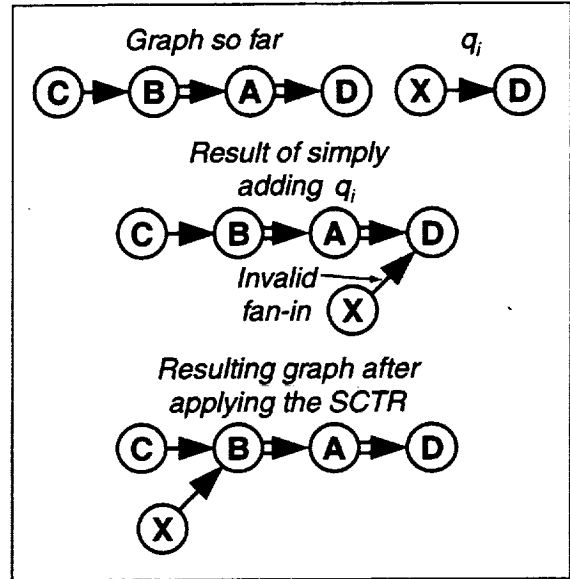
Figure 25. SCTR Applied to an Indirect Constraint

Case 4: The constraint q_i is a direct constraint, and either the SO ahead or the SO behind in the constraint (but not both) is already in the graph.

A new node is created for the SO in the constraint q_i that is not already in the graph. An edge is added that links the new node with the SO that is in q_i and already in the graph (see figure 26).
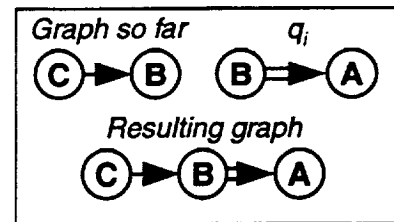
Figure 26. Direct Constraint Added to the Graph

Simply inserting q_i could exceed the fan-in/fan-out limit because of indirect constraints in the graph. In this case, the SCTR is applied, and the indirect constraints in the graph are moved to the end of the chain of direct constraints (see figure 27 and figure 28).
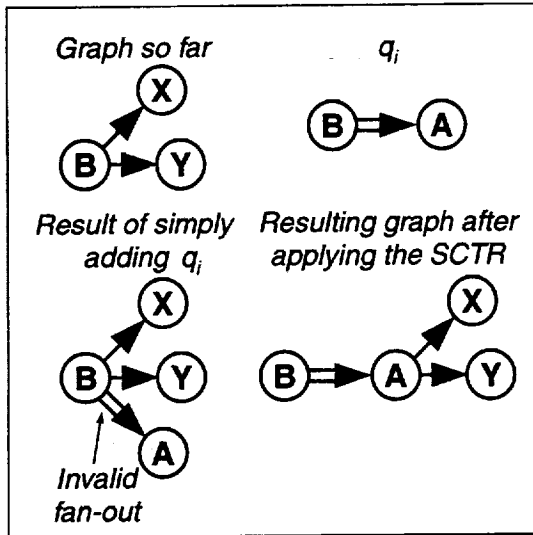
Figure 27. SCTR Applied to a Direct Constraint
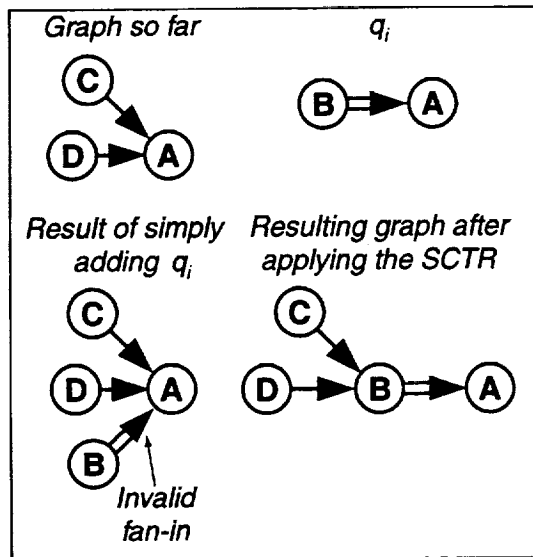
Figure 28. SCTR Applied to a Direct Constraint

Case 5: The constraint q_i is an indirect constraint, and both the SO ahead and the SO behind in the constraint are already in the graph.

No new nodes have to be created. The indirect constraint q_i is simply added to the graph (see figure 29).
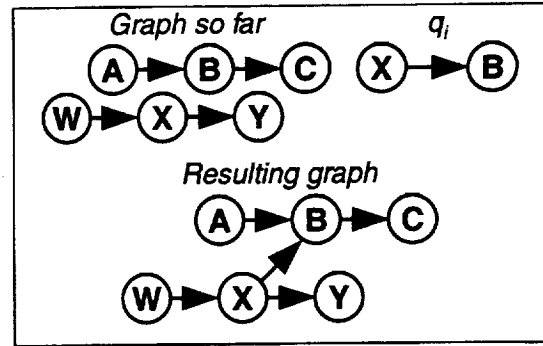
Figure 29. Indirect Constraint Added to the Graph

Simply inserting q_i could exceed the fan-in/fan-out limit because of direct constraints already in the graph. In this case, the SCTR is applied, and the indirect constraint q_i is moved to the end of the chain of direct constraints (see figure 30, figure 31, and figure 32).
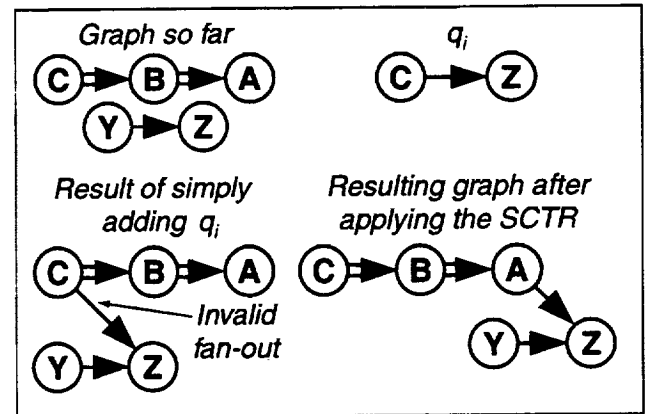
Figure 30. SCTR Applied to an Indirect Constraint
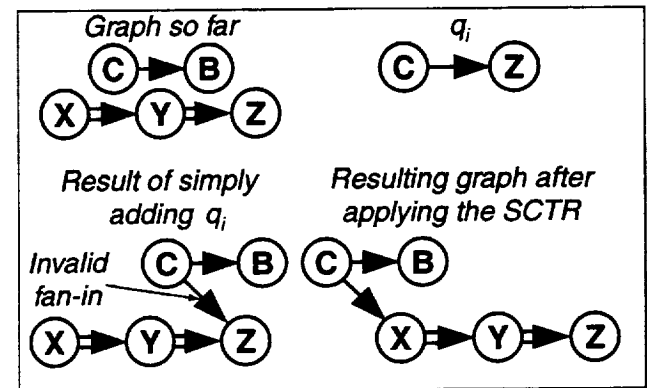
Figure 31. SCTR Applied to an Indirect Constraint
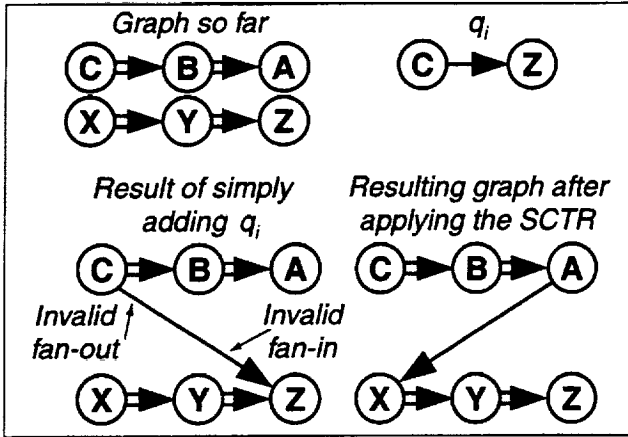
Figure 32. SCTR Applied to an Indirect Constraint

Case 6: The constraint q_i is a direct constraint, and both the SO ahead and the SO behind in the constraint are already in the graph.

No new nodes have to be created. In some cases, the constraint q_i can simply be inserted into the graph (see figure 33).
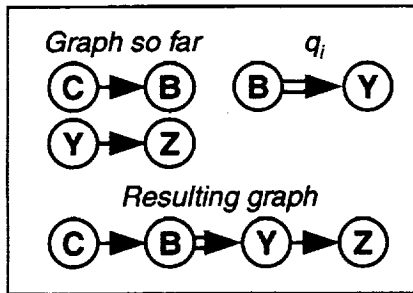
Figure 33. Adding a Direct Constraint to the Graph

Simply inserting q_i could exceed the fan-in/fan-out limit because of indirect constraints already in the graph. In these cases, the SCTR is applied after q_i is inserted, and the indirect constraints are moved to the end of the chain of direct constraints (see figure 34 and figure 35).
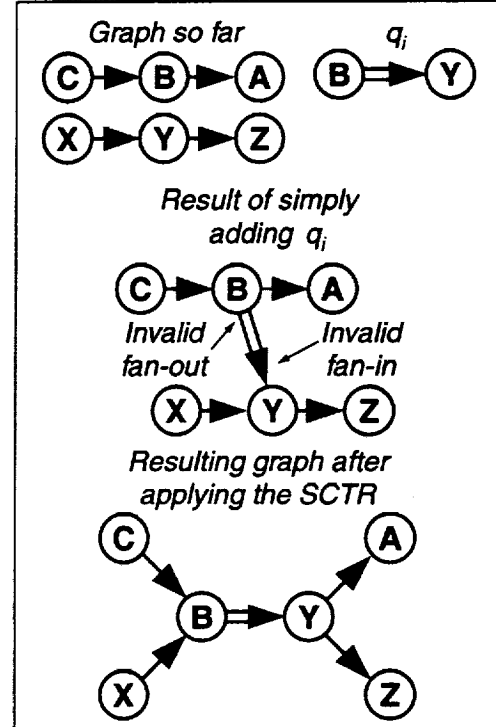
Figure 34. SCTR Applied to a Direct Constraint
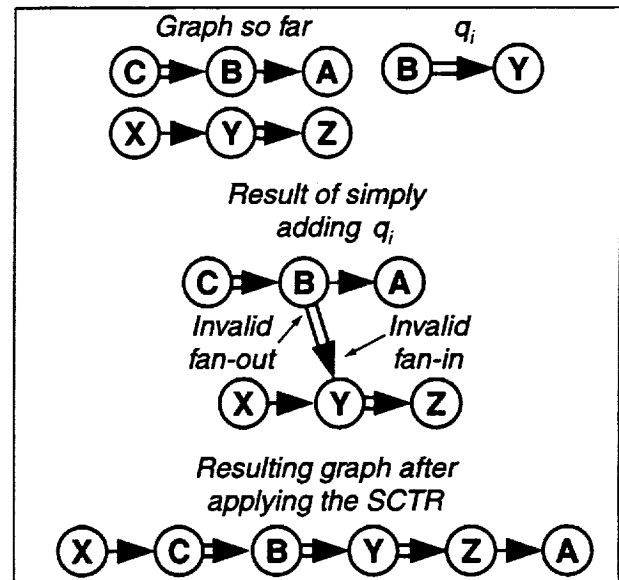
Figure 35. SCTR Applied to a Direct Constraint

3. Building the Combined Sequence to the Meter Fixes

All SOs are sequenced to their respective meter fixes and placed into one combined sequence. The scheduler can then extract sequencing information from each individual super stream class as required.

Step 1: Create previous sequence.

The sequencer begins by taking a list of SOs provided by the scheduler and places them into a sorted list that represents the previous sequence. The SOs are placed in this previous sequence, from earliest to latest, according to the meter fix STAs computed during the previous scheduling cycle. If two SOs have the same STA, then the tie is resolved using each SO's ETA to the meter fix. Some SOs don't have STAs because they were added to the system since the last scheduling cycle or changes have made their previous STA invalid. These SOs are placed at the end of the previous sequence after those SOs with the latest STAs.

Step 2: Deconflict manual sequence constraints.

The manual sequence constraints, those entered by the user, are deconflicted (see section II.H.2). The resulting Sequence Constraint Graph contains a consistent set of sequence constraints which will be applied in Step 7.

Step 3: Create a preliminary version of the new sequence.

The preliminary sequence represents the new sequence before any manually entered sequence constraints are applied. Each SO within the preliminary sequence has a flag which this document refers to as the "local-sequence-frozen" flag. This ad hoc flag is only used while constructing sequences and should not be confused with the SO's Sequence-Frozen indicator used throughout the DP. At the beginning of the sequencing process, the local-sequence-frozen flag for each SO is set to FALSE.

Step 4: Place Sequence-Frozen SOs into the preliminary sequence.

Next, if the scheduling mode requires that the sequence of Sequence-Frozen SOs is maintained, then SOs that are Sequence-Frozen are copied from the previous sequence (created in Step 1) and placed in a preliminary sequence (created in Step 3) in the same order. These Sequence-Frozen SOs have their local-sequence-frozen flag set to TRUE.

Step 5: Place Non-Sequence-Frozen SOs into the preliminary sequence.

For each of the remaining SOs, S_i, regardless of the scheduling mode, the following is carried out. S_i's meter fix ETA is compared against the meter fix ETA of each S_j already in the sequence. Exception: If S_j is manually scheduled, then S_i's meter fix ETA is compared against S_j's meter fix STA. S_i is inserted into the sequence after the last SO with a time (ETA or STA depending on the circumstances just described) that is equal to or earlier than S_i's meter fix ETA.

Step 6: Unfreeze SOs mentioned in a sequence constraint.

After the preliminary sequence has been built, each SO mentioned in the deconflicted sequence constraints has its local-sequence-frozen flag set to FALSE. This allows manual sequence constraints, when they are applied, to change the position of Sequence-Frozen SOs. Thus, at this point, local-sequence-frozen SOs are those that have their positions constrained only by being Sequence-Frozen and are not constrained by manual sequence constraints.

Step 7: Apply sequence constraints and create the final sequence.

In the final phase of sequencing, the sequence is modified to comply with the non-conflicting manual sequence constraints. The SOs are copied from the preliminary sequence and placed in the final sequence. This processing is done in the order that they should appear in the final sequence. Thus, when an SO is constrained behind one or more SOs, then it is not added to the end of the final sequence until all of the SOs constrained ahead of it have been added to the final sequence.

The details of this final step are as follows. First, the SOs in the preliminary sequence which meet the following criteria are examined.

The SO is not already in the final sequence, **and**

> The SO is not local-sequence-frozen, **or**
>
> The SO is local-sequence-frozen, and there are no local-sequence-frozen SOs ahead in the preliminary sequence, **or**
>
> The SO is local-sequence-frozen and the first local-sequence-frozen SO ahead in the preliminary sequence is already in the final sequence.

For each SO, S_i, which satisfies the preceding criteria, the DP's sequencer examines S_i to see if placing it at the end of the final sequence violates any sequence constraints. If S_i is constrained behind an SO which has not yet been placed in the final sequence, then there is a sequence constraint violation. On the other hand, if S_i is the preceding SO in a direct constraint, then the following SO in that constraint would have to be added to the end of the final sequence after S_i. This creates a dependency between S_i and the SO constrained directly behind. Thus, S_i violates a sequence constraint if the behind SO violates a sequence constraint. If the behind SO is itself the ahead SO in a direct sequence constraint, then it violates a sequence constraint if the other SO violates a sequence constraint. This continues recursively along the chain of direct sequence constraints.

Once it has been determined that adding S_j to the end of the final sequence does not violate any sequence constraints, S_j is added to the final sequence. If another SO is constrained directly behind S_j, then that other SO is added to the final sequence as well.

I. Scheduling

The goal of scheduling is to prepare a plan that delivers SOs from the Center to the TRACON in a smooth manner with minimal delay. This plan consists of STAs to various reference points, the outer meter arc, meter fix, FAF, and runway threshold for each SO. When scheduling SOs, each SO's ETA to each of the reference points represents the time of arrival if no other SOs were in the system. Thus, the ETA is used as the initial STA for each SO. A number of scheduling constraints are applied which delay the SO. This results in an STA that is farther in the future than the ETA. These scheduling constraints are entered by the TMC to reflect current procedures and conditions at the airports, in the TRACON, and in the Center. All of the scheduling constraints are applied, though some constraints may be more restrictive than others from SO to SO. The resulting schedule will consist of STAs for each SO that is as close to the SO's ETAs as possible while complying with all scheduling constraints. In addition, the scheduling process sets various status flags for each SO. The STAs and flags are then sent to the other CTAS processes.

1. Scheduling Events

The DP reschedules all or some of the SOs in response to various events. These events are listed in table 3 along with the scheduling modes (see section II.E.15) used when responding to these events.

Table 3. Scheduling Events

Scheduling Event Type	Scheduling Mode #	Immediate or Pending Event	Scheduling Event Time Reference Point
FOR_BLOCKED_INTERVAL_METER_FIX Addition or deletion of a meter fix blocked interval	5	Immediate	Meter Fix
FOR_FLOW_CHANGE_AIRPORT Change in the airport acceptance rate	5	Immediate	Runway
FOR_FLOW_CHANGE_RUNWAY Change in the occupancy time or required separation distance at a runway (This event is usually treated as a runway allocation event. However, if the runway allocator detects that a runway acceptance rate has not changed as a result of this event, then this event is treated as a scheduling event.)	5	Immediate	Runway
FOR_FLOW_CHANGE_GATE Change in the gate acceptance rate	5	Immediate	Meter Fix
FOR_FLOW_CHANGE_METER_FIX Change in the meter fix acceptance rate	5	Immediate	Meter Fix
FOR_FLOW_CHANGE_STREAM_CLASS Redefinition of super stream classes and/or change in the required separation distance at the meter fix	5	Immediate	Meter Fix
FOR_FLOW_CHANGE_TRACON Change in the TRACON acceptance rate	5	Immediate	Meter Fix

Table 3. Scheduling Events (Continued)

Scheduling Event Type	Scheduling Mode #	Immediate or Pending Event	Scheduling Event Time Reference Point
FOR_SEQUENCE_CONSTRAINT_AT_MF Addition of a meter fix sequence constraint	5	Immediate	Meter Fix
FOR_FIND_SLOT_AC - SO in Center's airspace A Find Slot operation initiated by the TMC	9	Immediate	Meter Fix
FOR_FIND_SLOT_AC - SO in TRACON's airspace A Find Slot operation initiated by the TMC	9	Immediate	Runway
FOR_MANUAL_SCHEDULE_SINGLE_AC_AT_RWY A runway STA manually set for an SO by the TMC	1	Immediate	Runway
FOR_MANUAL_SCHEDULE_SINGLE_AC_AT_MF A meter fix STA manually set for an SO by the TMC	1	Immediate	Meter Fix
FOR_FPA_FLOW_RUNWAY After receiving a flight plan amendment which changes an SO's runway, the first ETA to that runway is received	9	Immediate	Runway
FOR_FPA_COORD_FIX After receiving a flight plan amendment which changes an SO's coordination fix, the first corresponding ETA is received	9	Pending	Meter Fix
FOR_ETA_HOVER An SO's ETA is hovered	9	Pending	Meter Fix
FOR_FLOW_RUNWAY_ALLOCATION As part of the runway allocation process, the runway allocator will request that one or more schedules be computed	specified by allocator	Immediate - upon request from allocator	Runway
FOR_FLOW_RUNWAY_ALLOCATION_AC As part of the runway allocation processes, the runway allocator will request that one or more schedules be computed and the scheduling mode to be used	specified by allocator	Immediate - upon request from allocator	Runway
FOR_USER_REQUEST_AT_RWY TMC requests a reschedule via a TGUI runway timeline menu	specified in request	Immediate	Runway
FOR_USER_REQUEST_AT_FAF TMC requests a reschedule via a TGUI final approach fix timeline menu	specified in request	Immediate	All SOs requested: Runway Otherwise: FAF

Table 3. Scheduling Events (Continued)

Scheduling Event Type	Scheduling Mode #	Immediate or Pending Event	Scheduling Event Time Reference Point
FOR_USER_REQUEST_AT_MF TMC requests a reschedule via a meter fix timeline menu	specified in request	Immediate	All SOs requested: Runway Otherwise: Meter Fix
FOR_PERIODIC_RESCHEDULE At least 6 seconds since a reschedule involving an even-numbered mode .	8	Immediate	Runway

The scheduling events are further broken down into immediate events and pending events. Immediate events will trigger a reschedule at once while pending events are placed in a pending list for rescheduling later. This deferred scheduling actually takes place when one of the following occurs:

1. An immediate scheduling event is received.

2. A pending event is received that is not the same type as events stored in the pending list. Each event listed in table 3 is of a separate type.

3. A pending event is received and the pending list is full.

Since all of the events in the pending list are of the same type (see item 2 above), processing the entire list of pending scheduling events requires only a single schedule computation. Once the events in the pending list have been processed, the pending list is cleared.

2. Order of Consideration at the Runway

After the SOs have been scheduled to the meter fixes, they are scheduled to the runway. Because a sequence generated at the runway could contradict the sequence at the meter fix, sequences at the runway are not computed, and sequence constraints relative to the runway are disallowed. Instead, an Order of Consideration at the runway is computed by the Scheduler class.

The Order of Consideration is the order in which SOs have their runway threshold and FAF STAs computed. This does not necessarily mean that the SOs will be scheduled to arrive at the runway in this order, but only that they are computed in this order. Thus, the scheduler will have greater latitude in computing the STAs for SOs which are earlier in the order versus SOs which are later in the order. The sequence at the meter fix will remain unchanged by the Order of Consideration algorithm.

The Order of Consideration algorithm is executed when runway STAs are computed. It assumes that a preliminary meter fix STA has already been computed for each SO. The algorithm begins by determining the SO with the earliest meter fix STA within each super stream class. Among these SOs, the SO with the earliest runway ETA is selected as the next SO in the order of consideration. Next, this SO has its runway threshold and FAF STAs computed and any necessary delay is fed back to its meter fix STA. This algorithm is repeated until all SOs have been scheduled to the runway.

Consider the example in table 4. Aircraft A1 and A2 are in stream class A. B1 and B2 are in stream class B. C1 and C2 are in stream class C. The table also shows the preliminary meter fix STA and the runway ETA for each of these aircraft. For this example, assume that aircraft must maintain 1 minute separation at the meter fix from other aircraft within its stream class. Also, assume that the required separation at the runway is 1.5 minutes.

Table 4. Order of Consideration Example

ID	Stream Class	Prelim. MF STA (Zulu)	Runway ETA (Zulu)	Computed Runway STA (Zulu)
A1	A	12:00:00	12:12:00	12:12:00
A2	A	12:02:00	12:14:00	12:15:00
B1	B	12:00:00	12:15:00	12:16:30
B2	B	12:01:00	12:16:00	12:18:00
C1	C	12:00:00	12:10:00	12:10:00
C2	C	12:03:00	12:13:00	12:13:30

The Order of Consideration algorithm determines that A1, B1, and C1 have the earliest meter fix STA for each stream class. Among these, C1 has the earliest runway ETA, so its runway STA is computed first. C1's runway STA is computed to be 12:10:00Z.

Now that C1 has been scheduled, the algorithm determines that A1, B1, and C2 have the earliest meter fix STA for each stream class. Since A1 has the earliest runway ETA, it is next in the Order of Consideration. A1's runway ETA differs from C2's runway STA by more than the required 1.5 minutes. Therefore, A1 is not delayed, and it is scheduled at its ETA, 12:12:00Z.

The next set of aircraft with the earliest meter fix STA within each stream class consists of A2, B1, and C2. In this case, C2 has the earliest runway ETA, so it is next in the Order of Consideration. In order to maintain a 1.5 minute separation from A1, C2 is delayed slightly and scheduled at 12:13:30Z.

Of the remaining aircraft, A2 and B1 have the earliest meter fix STAs within their respective stream classes. Since A2 has the earlier runway ETA, it is next in the Order of Consideration. The only slot available to A2 is 1.5 minutes behind C2. Thus, A2 is delayed at the runway by a minute and is scheduled to land at 12:15:00Z.

Finally, B1 and then B2 are scheduled to the runway. Because of the other aircraft already scheduled, B1 and B2 will be delayed to satisfy the 1.5 minute separation requirement at the runway.

3. Delay Feedback

Due to scheduling constraints at the airport and its runways, an SO may be delayed at the runway threshold. The measure of this delay is based on the sum of the meter fix STA (STA_{mf}) and the TRACON Transition Time ($T_{mf\text{->}rwy}$). The TRACON Transition Time is defined as the time required to fly from the meter fix to the runway threshold when there are no other SOs present. Thus,

$$T_{mf\text{->}rwy} = ETA_{rwy} - ETA_{mf} \qquad (1)$$

where ETA_{rwy} is the ETA to the runway threshold and ETA_{mf} is the ETA to the meter fix.

In addition, scheduling constraints may cause delays at the meter fix. Such delays make it impossible for an SO to meet the runway ETA. The earliest time that an SO can arrive at the runway threshold is its Proposed Time of Arrival (PTA_{rwy}) and is computed according to:

$$PTA_{rwy} = STA_{mf} + T_{mf\text{->}rwy} \qquad (2)$$

Finally, it follows that the amount of delay at the runway threshold is:

$$Delay_{rwy} = STA_{rwy} - PTA_{rwy} \qquad (3)$$

To optimize performance, this delay must be distributed between the Center and the TRACON. Adding delay to the meter fix STA as a result of excessive delay at the runway threshold is handled in the DP by a mechanism known as Delay Feedback.

After a preliminary STA is computed to the meter fix and the runway threshold for a particular SO, the delay at the runway is examined. The maximum amount of delay that can be absorbed in the TRACON is a parameter of the DP. If the runway delay is within the amount that can be absorbed in the TRACON, then no adjustments to the SO's STAs are necessary. However, if the amount of delay is greater than the amount that can be absorbed in the TRACON, then the excess delay is fed back or added to the meter fix STA. This delays the SO at the meter fix. If delaying the SO at the meter fix results in a violation of a scheduling constraint, then the process of computing the SO's STA is repeated. The times that were just computed serve as the earliest allowed times of arrival during this next scheduling iteration. This is necessary because any meter fix delay beyond that required by the delay feedback could make the runway threshold STA unrealizable.

A similar mechanism is used by the DP to distribute the delay between the low altitude arrival sector and the high altitude sector. The SO's meter fix STA is used by the low altitude arrival sector while the outer meter arc STA is used by the high altitude arrival sector. No scheduling constraints are applied at the outer meter arc. The Outer Meter Arc to Meter Fix Transition Time ($T_{oma\text{->}mf}$) is the time it takes an SO to fly from the outer meter arc to the meter fix when there are no other aircraft in the airspace.

$$T_{oma\text{->}mf} = ETA_{mf} - ETA_{oma} \qquad (4)$$

Once the meter fix STA (STA_{mf}) is computed, the outer meter arc STA (STA_{oma}) is computed according to:

$$STA_{oma} = STA_{mf} - T_{oma\text{->}mf} - AMDT \qquad (5)$$

where AMDT is the Amount of Delay Time. Without the AMDT, the high altitude arrival sector would be required to absorb all Center delays. By subtracting the AMDT, the low altitude arrival sector is forced to absorb up to the amount represented by the AMDT. The AMDT is a site-dependent parameter of the DP.

23

4. Scheduling Process

The scheduling events described in section II.I.1 trigger the scheduling process. In addition, the Runway Allocator will request the computation of schedules for each runway assignment it is considering for an SO. The Runway Allocator compares these schedules to determine which runway assignment is best. In either case, the scheduling process follows the steps summarized in figure 36. The numbers shown in figure 36 correspond to the step numbers given in this section's description.
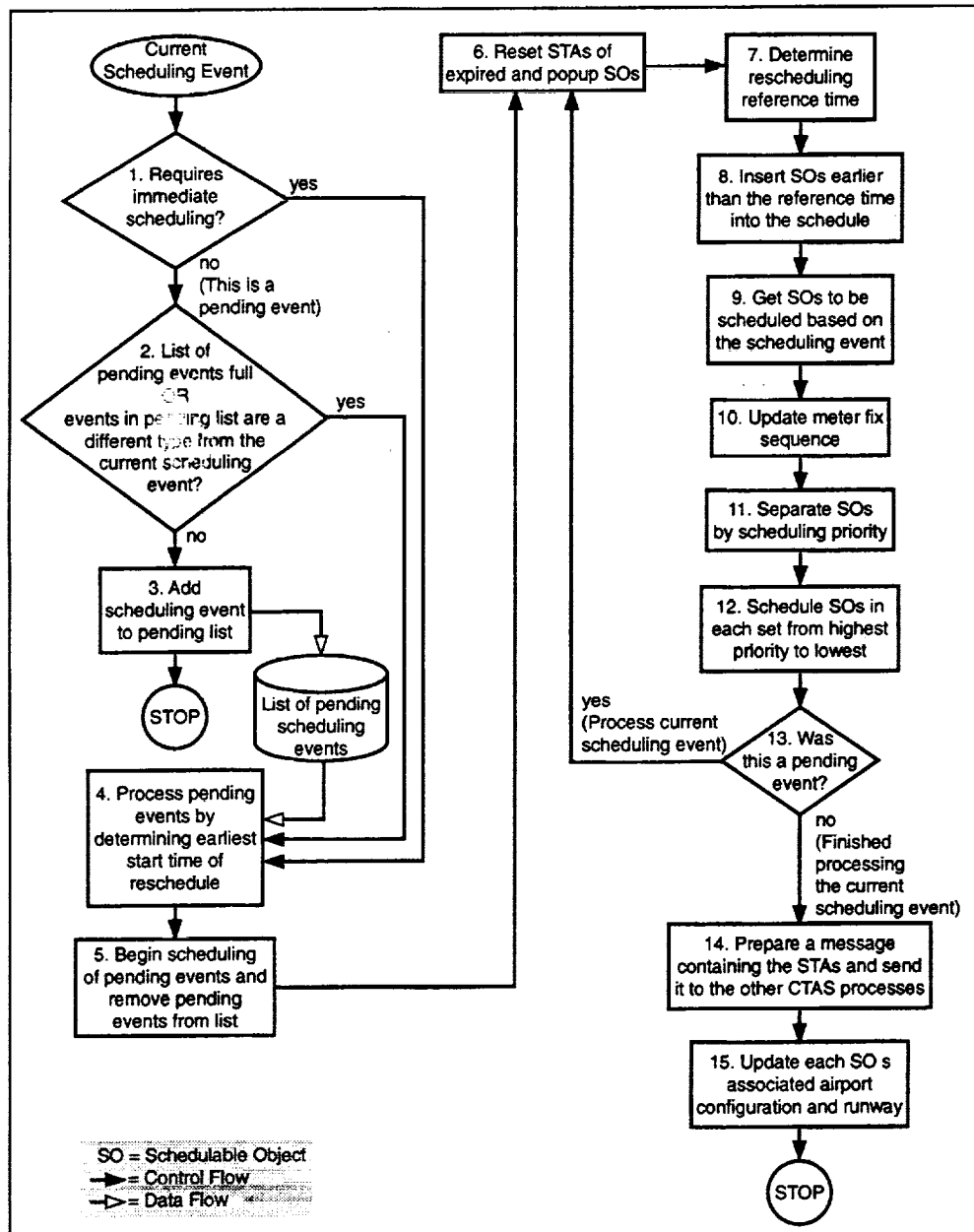

Figure 36. Top Level Scheduling Flow Chart

Step 1: Determine if the current scheduling event requires immediate rescheduling (see section II.I.1 and table 3). If this is an immediate rescheduling event, then go to Step 4.

Step 2: The current scheduling event is a pending event. If the list of pending scheduling events is full or the events in the list are different from this rescheduling event, then go to Step 4.

Step 3: Add this event to the list of pending scheduling events and quit.

Reaching this step means that the current scheduling event is a pending event, there is room in the pending list, and this event is of the same type as those in the pending list. This event is added to the pending list, and the scheduling process is ended at this point. This event will be processed later, along with the other pending events in Step 4 in response to a different scheduling event.

Step 4: Begin processing the events in the list of pending scheduling events (if any) by determining the earliest start time for the reschedule.

All of the pending scheduling events in the list are of the same type. They use the same scheduling mode and method. Therefore, only one reschedule is necessary for the entire set of events. To ensure that all events are covered, each event is examined to find the earliest reschedule start time necessary.

Step 5: Carry out the scheduling process for the pending event starting with Step 6, and remove the events from the list of pending scheduling events.

Scheduling triggered by a pending event is handled in a manner similar to scheduling triggered by an immediate event. The start time of the reschedule is that time found in Step 4 above. Since a single rescheduling process is enough to handle all of the pending scheduling events in the list, the list can be cleared after the reschedule is complete.

Step 6: Reset the STAs of all expired and popup SOs.

Since the last schedule was computed, some SOs may have been flagged as expired (see section II.F.2) or as pop-ups (see section II.F.3). Resetting their STAs ensures that any previous STAs are not stored in the CM nor sent to the Center's Host computer. Additionally, the STAs of these SOs are not displayed on any of the GUIs.

Step 7: Determine the point in time where rescheduling is to be executed.

Not all SOs are affected by the events triggering rescheduling. In most cases, SOs due to arrive later are usually influenced by the SOs due to arrive sooner. In

contrast, SOs due to arrive sooner are usually not influenced by the SOs due to arrive later in the absence of sequence constraints. However, sequence constraints can cause some SOs to influence the schedule of SOs that are due to arrive earlier.

A rescheduling start time must be derived from the scheduling event. SOs whose ETAs or STAs are equal to or later than the rescheduling start time are rescheduled. The rescheduling start time is equal to the Scheduling Event Time minus the Zone of Influence. The Scheduling Event Time and the Zone of Influence are described below.

The rescheduling start time may be relative to the meter fix, FAF, or runway. For example, adding a meter fix acceptance rate means that the rescheduling start time must be relative to the meter fix. The DP determines from the event which reference point to use when determining which SOs are to be rescheduled (see table 3).

Scheduling Event Time. Some scheduling events have an explicit Scheduling Event Time associated with them. For example, the user may enter a scheduling constraint and the time when the scheduling constraint should be active. This activation time is the Scheduling Event Time.

Other scheduling events have a Scheduling Event Time which is based on an SO's ETA. These are scheduling events that have a specific SO associated with them. For example, the user may request that rescheduling be executed for a particular SO and all SOs following it. In such a case, the Scheduling Event Time is the ETA of the SO.

Zone of Influence. It is not sufficient to set the Rescheduling Start Time equal to the Scheduling Event Time. The nature of some scheduling constraints makes it necessary to reschedule SOs that are due to arrive slightly earlier than the Scheduling Event Time.

The Zone of Influence is the mechanism by which the DP accounts for the rescheduling of SOs due to arrive earlier than the Scheduling Event Time. The Zone of Influence is set based on the maximum of the Acceptance Rate Interval (see section II.E.3) and the sum of the largest occupancy time constraint and largest separation distance constraint (see section II.I.10) active at the Scheduling Event Time.

25

Step 8: Get all SOs with ETAs that are earlier than the rescheduling start time (see Step 7) and insert them into the schedule based on their previously computed STAs.

SOs earlier than the rescheduling time are unaffected by the current scheduling event, so their STAs are left unchanged. However, they must be included in the schedule since they may affect the STAs of those SOs that are being rescheduled.

Step 9: Get all SOs that are to be rescheduled as a result of this scheduling event.

For each SO, a series of conditions are checked to determine if the SO should be rescheduled. These conditions are summarized in a truth table (see table 5). If the SO is eligible for rescheduling, it is placed in a list that is used in the following steps.

For example, the fifth column from the right in table 5 shows that an SO is eligible for rescheduling if the following conditions are true:

- The SO has valid ETAs.
- The SO's ETA is **not** earlier than the rescheduling start time.
- The SO has **not** landed.
- The SO has **not** been suspended.
- The SO is **not** a pop-up.
- The SO is **not** expired.
- The SO is active.

The "*" indicates that if the above conditions are true, then it does not matter if the SO is a proposed flight plan or not, and it does not matter if the SO has departed or not.

Table 5. Scheduling Eligibility Truth Table

SO Status												
has valid ETAs	0	*	*	*	*	*	*	1	1	1	1	1
ETA is earlier than the rescheduling start time	*a	1	*	*	*	*	*	0	0	0	0	0
landed	*	*	1	*	*	*	*	0	0	0	0	0
suspended	*	*	*	1	*	*	*	0	0	0	0	0
pop-up	*	*	*	*	1	*	*	0	0	0	0	0
expired	*	*	*	*	*	1	*	0	0	0	0	0
proposed flight plan	*	*	*	*	*	*	*	*	0	1	0	1
departed (User manually departed this SO.)	*	*	*	*	*	*	*	*	0	0	1	1
active (DP has received radar track data for this SO.)	*	*	*	*	*	*	0	1	0	0	0	0
schedule flight plans (The DP has been set to include inactive SOs in the scheduling process. This is normally switched on.)	*	*	*	*	*	*	0	*	1	1	1	1
Eligible for Scheduling	0	0	0	0	0	0	0	1	1	0	1	1

a. * = any value

26

Step 10: Update the sequence to the meter fix of the SOs that are being rescheduled.

The SOs selected for rescheduling are sorted into a sequence to the meter fix. The sequence is based on an FCFS order based on meter fix ETA. The sequence is further refined so that SOs which are Sequence-Frozen maintain their sequence relative to each other as dictated by the scheduling mode (see section II.E.15). The sequence is also constructed so that any sequence constraints received by the DP are also observed. For a detailed description of the sequencing process, see section II.H.3.

Step 11: The SOs are broken down into sets that have different scheduling priorities.

The SOs are broken down into Scheduling Priority Sets. Each set has a scheduling method associated with it. The definitions of these sets and their corresponding scheduling methods vary depending on the scheduling mode (see table 6, table 7, and table 8). Each scheduling method is detailed in its own section later in this paper.

The SOs with a higher priority are scheduled first and have the best chance of actually being scheduled at their ETAs. The SOs with a lower scheduling priority are scheduled around the SOs with a higher scheduling priority. Each table lists the Scheduling Priority Sets in order from highest priority to lowest.

Step 12: Each set of SOs is processed one at a time from the highest priority to the lowest priority.

Each SO within each set is inserted into the new schedule using one of three methods depending on the scheduling mode and the set's scheduling priority. Each method is summarized below and is described in its own section.

Insert without Rescheduling. This method inserts the SO into the schedule based on the STA computed in a previous scheduling cycle (see section II.I.5).

Reschedule at ETA and Insert. This method is usually applied when scheduling priority aircraft. The SO is scheduled at its ETA and is only delayed to avoid a conflict with an SO that has already been scheduled such as another priority aircraft or an STA-Frozen SO. Sequence is not considered by this method (see section II.I.6).

Reschedule after Aircraft Ahead and Insert. The bulk of the SOs are scheduled using this method. The sequence computed in Step 10 is used when computing the STA of an SO under this method (see section II.I.7).

Each method, except Insert without Rescheduling, executes the following steps:

1. Compute STAs to the meter fix in sequence.

2. Compute STAs to the runway in order of consideration.

3. Adjust meter fix STAs to account for delay feedback.

4. Compute outer meter arc STAs.

In addition, all three methods create a Schedule Linked List data structure. This list is used to identify which SOs have already been scheduled and can affect the STA calculation of the SO currently being scheduled.

Another data structure used during scheduling is the set of Acceptance Rate Bins. The Acceptance Rate Bins track the number of SOs which have been scheduled in a particular period of time. This data structure is used when applying an acceptance rate scheduling constraint to the SO currently being scheduled. For more details on Acceptance Rate Bins, see section II.I.11.

Once the current SO has been scheduled, its STA determines where the SO should be inserted into each of these data structures.

Step 13: If this scheduling activity is part of the processing of the list of pending scheduling events (see Step 5), then the processing of the pending scheduling events is complete. Begin processing the current scheduling event starting at Step 6. However, if this scheduling activity is part of the processing of the current scheduling event, then proceed to the next step.

Step 14: Prepare the scheduling message containing the newly computed STAs and other SO status indicators, and send the message to the rest of CTAS.

To minimize the amount of message traffic over the network, not all SOs are contained in the scheduling message. SOs which have a valid STA are included in the message. In addition, SOs which have had certain flags set by the DP are included in the message even if no STAs have been computed for them. These special flags are examined by the GUIs and affect the display of the applicable SOs.

Step 15: Update each SO's associated airport configuration and runway.

An SO may be delayed enough to cause it to be associated with a different airport configuration than it was prior to scheduling. In this case, the DP assigns the SO to the default runway of the new airport configuration. For each SO which has had its associated airport configuration changed or both its associated airport configuration and assigned runway changed, a

flight plan amendment is sent notifying the rest of CTAS of the change.

Table 6. Schedule All Including STA-Frozen SOs (modes 2, 3, 4, and 5)

Scheduling Priority Set	Scheduling Method
SOs with ETAs earlier than the rescheduling start time	Insert without Rescheduling
Manually scheduled SOs	Insert without Rescheduling
Priority SOs	Reschedule at ETA and Insert
Other SOs	Reschedule after Aircraft Ahead and Insert

Table 7. Schedule All Non-STA-Frozen SOs (modes 1, 6, 7, 8, and 9)

Scheduling Priority Set	Scheduling Method
SOs with ETAs earlier than the rescheduling start time	Insert without Rescheduling
Manually scheduled SOs	Insert without Rescheduling
STA-Frozen priority SOs	Insert without Rescheduling
STA-Frozen non-priority SOs	Insert without Rescheduling
Non-STA-Frozen priority SOs	Reschedule at ETA and Insert
Other SOs	Reschedule after Aircraft Ahead and Insert

Table 8. Schedule All Non-Sequence-Frozen SOs (modes 10 and 11)

Scheduling Priority Set	Scheduling Method
SOs with ETAs earlier than the rescheduling start time	Insert without Rescheduling
Sequence-Frozen SOs	Insert without Rescheduling
Non-Sequence-Frozen priority SOs	Reschedule at ETA and Insert
Other SOs	Reschedule after Aircraft Ahead and Insert

5. Insert without Rescheduling

This method is used when an SO is not to have its STA changed by the scheduling process. The SO is simply inserted into the Schedule Linked List and Acceptance Rate Bins based on the STA computed during a previous scheduling update. Even though its STA is not changed, including the SO in the scheduling data structures ensures that its influence on the STAs of other SOs is taken into account.

6. Reschedule at ETA and Insert

This method is used for scheduling priority SOs. Although a sequence is computed, the sequence is used only as the order in which the SOs are processed. An attempt is made to schedule each SO at its ETA disregarding the sequence. If scheduling the SO at the ETA would cause it to violate a scheduling constraint, then the SO is delayed an amount sufficient to comply with all scheduling constraints.

The steps executed under this method are summarized in figure 37. The numbers included in figure 37 correspond to the step numbers listed in the following description.

28
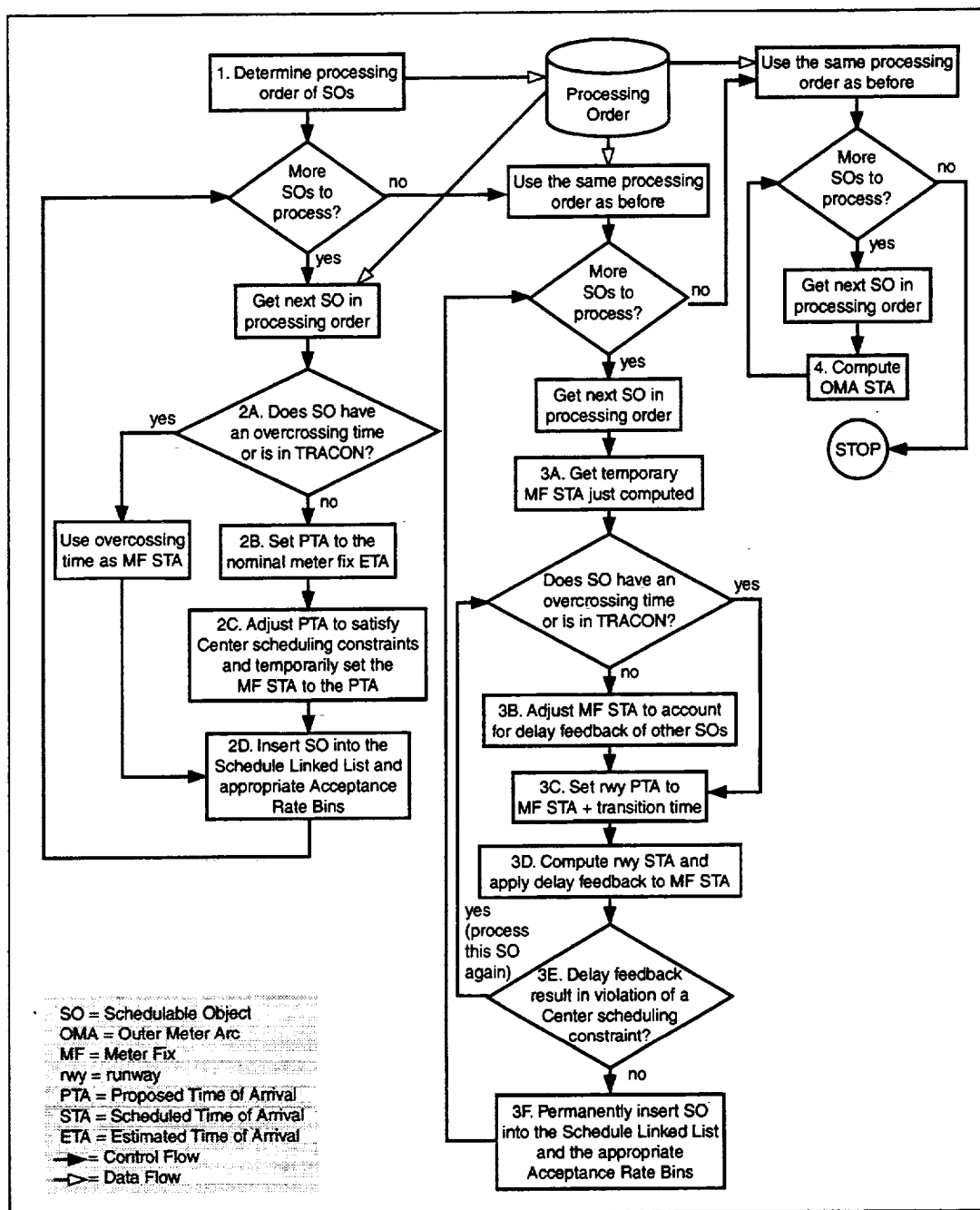
Figure 37. Reschedule at ETA and Insert Flow Chart

Step 1: Determine the processing order of the SOs by determining the sequence at the meter fix (see section II.H.3).

This method does not actually obey the meter fix sequence when computing the STAs for the SOs in the set. The sequence is simply used as the order in which SOs are processed by this method.

Step 2: For each SO to be scheduled using this method, Step 2A through Step 2D are executed.

Step 2A: If an overcrossing time has been received for this SO or the SO is completely within the TRACON's airspace, then the meter fix STA is set to the overcrossing time. Go to Step 2D.

The overcrossing time (see section II.B.4) is used as the meter fix STA and is not adjusted to comply with Center scheduling constraints since these constraints no longer

29

have any effect. Note that it is necessary to compute the meter fix STA for SOs that have already crossed over the meter fix because STAs in the past have an impact on present and future STAs via the acceptance rate constraints. For example, suppose that a particular meter fix has an acceptance rate of 24 aircraft per hour, and 20 aircraft have already crossed over the meter fix in the past 50 minutes. The scheduling constraint should prevent more than 4 aircraft from crossing the meter fix in the next 10 minutes. Only by computing the past meter fix STAs of those 20 aircraft using their overcrossing times can this limitation of 4 aircraft be properly enforced.

Step 2B: If the conditions in the previous step are not true, then get the meter fix ETA. This becomes the Proposed Time of Arrival (PTA).

The PTA represents a temporary STA until the final STA is computed. Since the meter fix ETA is the earliest possible time that an SO can be scheduled to arrive at the meter fix, the SO's PTA is initialized to the SO's ETA. As scheduling constraints are applied, the SO's PTA will be delayed.

Step 2C: Modify the PTA to satisfy all Center scheduling constraints. Use the modified PTA as the meter fix STA.

Each of the Center scheduling constraints is applied to the PTA possibly pushing the PTA later. The more restrictive scheduling constraints will have a greater impact on a particular SO, and which constraint is more restrictive will vary from SO to SO. The following is a list of scheduling constraints which affect the computation of the meter fix STA.

- TRACON acceptance rate
- Meter fix acceptance rate
- Gate acceptance rate
- Super stream class separation Miles-in-Trail
- Meter fix blocked intervals

Step 2D: If an STA was successfully computed from Step 2A or Step 2C, then insert the current SO into the Schedule Linked List and the appropriate Acceptance Rate Bins.

The meter fix STA just computed is stored in the Schedule Linked List and Acceptance Rate Bins as a temporary meter fix STA. It is used to detect scheduling conflicts between this SO and any other SOs which are subsequently processed in Step 2. However, since this is a temporary meter fix STA assignment, other SOs, which are being scheduled to the runway threshold and have their delay fed back to the Center, ignore this SO's meter fix STA when applying the scheduling constraints.

Eventually, when this SO has had its runway STA computed, and its meter fix STA has been adjusted for delay feedback, its meter fix STA will be permanently assigned. A permanently assigned STA will impact the STAs of other SOs as they are being scheduled to the runway.

Step 3: For each SO to be scheduled using this method, Step 3A through Step 3F are executed.

The sequence computed in Step 1 is used as the processing order when computing the runway STAs.

Step 3A: Get the temporary meter fix STA just computed in Step 2.

Step 3B: If an overcrossing time has not been received for this SO and the SO is not completely within the TRACON's airspace, adjust the meter fix STA to account for SOs that have had some delay fed back to the Center as a result of executing Step 3 on those other SOs.

If the current SO has an overcrossing time or is completely within the TRACON's airspace, then use the overcrossing time as the meter fix STA. No adjustment of this meter fix STA is necessary since the SO has already crossed the meter fix.

If, however, this SO does not have an overcrossing time and is in the Center's airspace, then the meter fix STA may require adjustment so that it satisfies the scheduling constraints at the meter fix. The meter fix STA of the current SO is only compared against those SOs with permanently assigned meter fix STAs. In other words, the current SO is compared against those SOs which have already had their runway STAs computed and their delay fed back to the Center. If the meter fix STA is modified, then adjustments are made to the Schedule Linked List and the Acceptance Rate Bins to correspond to the modified meter fix STA.

Step 3C: Set the runway PTA to the sum of the meter fix STA and the transition time from the meter fix to the FAF or runway threshold (see equation 1 and equation 2).

If the SO does not have a meter fix STA (usually because the SO is completely within the TRACON's airspace and no overcrossing time was received), then the runway PTA is set to the runway ETA. Otherwise the value that equation 2 yields is used as the runway PTA.

Whether to use the FAF or the threshold depends on the airport's configuration (see section II.E.8).

Step 3D: Compute the SO's runway STA and apply any delay feedback to its meter fix STA.

This step is described later. For details, see section II.I.8.

Step 3E: If delay feedback caused the current SO to violate a Center scheduling constraint, then recompute the meter fix and runway STAs by going back to Step 3A.

As explained in section II.I.3, if delay feedback puts the current SO's meter fix STA in violation of a scheduling constraint, then the SO's meter fix STA must be recomputed. This is because it will be necessary to delay the meter fix STA further to avoid violating the scheduling constraint, and this new meter fix STA will make the runway STA computed in Step 3D impossible to meet. The meter fix STA plus delay feedback, after adjustment for scheduling constraints, is used as the meter fix STA when the process returns to Step 3A with the current SO.

Step 3F: If delay feedback did not cause the current SO to violate any Center scheduling constraints, then permanently insert the SO into the Schedule Link List and Acceptance Rate Bins.

At this point, the current SO has been successfully scheduled to the runway threshold, FAF, and meter fix. The current SO is permanently placed in the Schedule Link List and Acceptance Rate Bins. This means that the current SO can affect the schedules of all remaining SOs that need to be scheduled.

Step 4: For each SO to be scheduled using this method, compute the outer meter arc STA using equation 5.

7. Reschedule after Aircraft Ahead and Insert

This method is used for most of the common cases of scheduling. The sequence computed in Step 10, section II.I.4 is followed when computing the meter fix STAs and maintained when computing the runway STAs. Note that SOs in different super stream classes are sequenced independently of each other. Thus, there is no restriction as to the order of SOs from different super stream classes relative to each other.

Once the preliminary meter fix STAs have been computed, the runway STA for each SO is computed. There is no rigid sequence to the runway because such a sequence might contradict the sequence at the meter fix. Therefore, the DP constructs an Order of Consideration (see section II.I.2). This is the order in which SOs are considered for the available slots at the runway. Under most circumstances, the sequence at the runway will correspond to the Order of Consideration. However, the freedom exists to allow the sequences at the runways to differ from the Order of Consideration in order to maintain the sequence at the meter fix.

The steps executed by this method are summarized in figure 38. The numbers included in figure 38 correspond to the step numbers listed in the following description.

Step 1: For each SO, in the order of the meter fix sequence, Step 1A through Step 1D are executed.

Step 1A: If an overcrossing time has been received for this SO or the SO is completely within the TRACON's airspace, then the meter fix STA is set to the overcrossing time. Go to Step 1D.

This is the same as Step 2A of section II.I.6.

Step 1B: If the conditions in the previous step are not true, then set the PTA to be the later of the nominal meter fix ETA and the meter fix STA of the SO ahead of the current SO in the sequence and in the same super stream class.

The meter fix ETA is used as the earliest possible time that an SO can be scheduled to arrive at the meter fix if there are no other aircraft to consider. Additionally, in order to maintain the sequence at the meter fix, the current SO cannot be scheduled earlier than the SO just ahead of it in the same super stream class. Thus, the PTA is set to the later of either the nominal meter fix ETA or the STA of the SO ahead of the current SO which is in the same super stream class. Scheduling constraints and delay feedback may delay this aircraft so that its meter fix STA is even later than the time computed in this step.

Step 1C: Adjust the PTA to satisfy all Center scheduling constraints. Use the adjusted PTA as the meter fix STA.

This is the same as Step 2C of section II.I.6.

Step 1D: If an STA was successfully computed in Step 1A or Step 1C, then insert the current SO into the Schedule Linked List and the appropriate Acceptance Rate Bins.

Step 2: Schedule SOs to the runway in Order of Consideration (see section II.I.2). For each SO to be scheduled to the runway, Step 2A through Step 2I are executed.

Step 2A: Of the SOs yet to be scheduled at the runway, find the one with the earliest meter fix STA.

The meter fix STA of the earliest SO is used in the following step when determining the super stream class of each SO. The grouping of stream classes into super stream classes can vary over time if future super stream class scheduling constraints have been entered by the TMC. Therefore, the meter fix STA of the earliest SO is used for the sake of determining which SO should be considered next for runway scheduling.

Get MF Sequence → MF Sequence

MF Sequence (data store)

Use the same processing order as before

More SOs to process?
- no →
- yes ↓

Get next SO in sequence

1A. Does SO have an overcrossing time or is in TRACON?
- yes → Use overcrossing time as MF STA
- no ↓

1B. Set PTA to be the later of the nominal MF ETA or the MF STA of the SO ahead in the super stream class

1C. Adjust PTA to satisfy Center scheduling constraints and temporarily set the MF STA to the PTA

1D. Insert SO into the Schedule Linked List and appropriate Acceptance Rate Bins

More SOs to schedule to rwy?
- no →
- yes ↓

2A. Find the SO with the earliest MF STA

2B. Determine the SO with the earliest MF STA within each super stream class

2C. Of the earliest SO within each super stream class, select the SO with the earliest rwy ETA

2D. Get the temporary MF STA just computed

Does SO have an overcrossing time or is in TRACON?
- yes →
- no ↓

2E. Adjust MF STA to account for delay feedback of other SOs

2F. Set rwy PTA to MF STA + transition time

2G. Compute rwy STA and apply delay feedback to MF STA

2H. Delay feedback result in violation of a Center scheduling constraint?
- yes (process this SO again)
- no ↓

2I. Permanently insert SO into the Schedule Linked List and the appropriate Acceptance Rate Bins

More SOs to process?
- no →
- yes ↓

Get next SO in processing order

3. Compute OMA STA

STOP

SO = Schedulable Object
OMA = Outer Meter Arc
MF = Meter Fix
rwy = runway
PTA = Proposed Time of Arrival
STA = Scheduled Time of Arrival
ETA = Estimated Time of Arrival
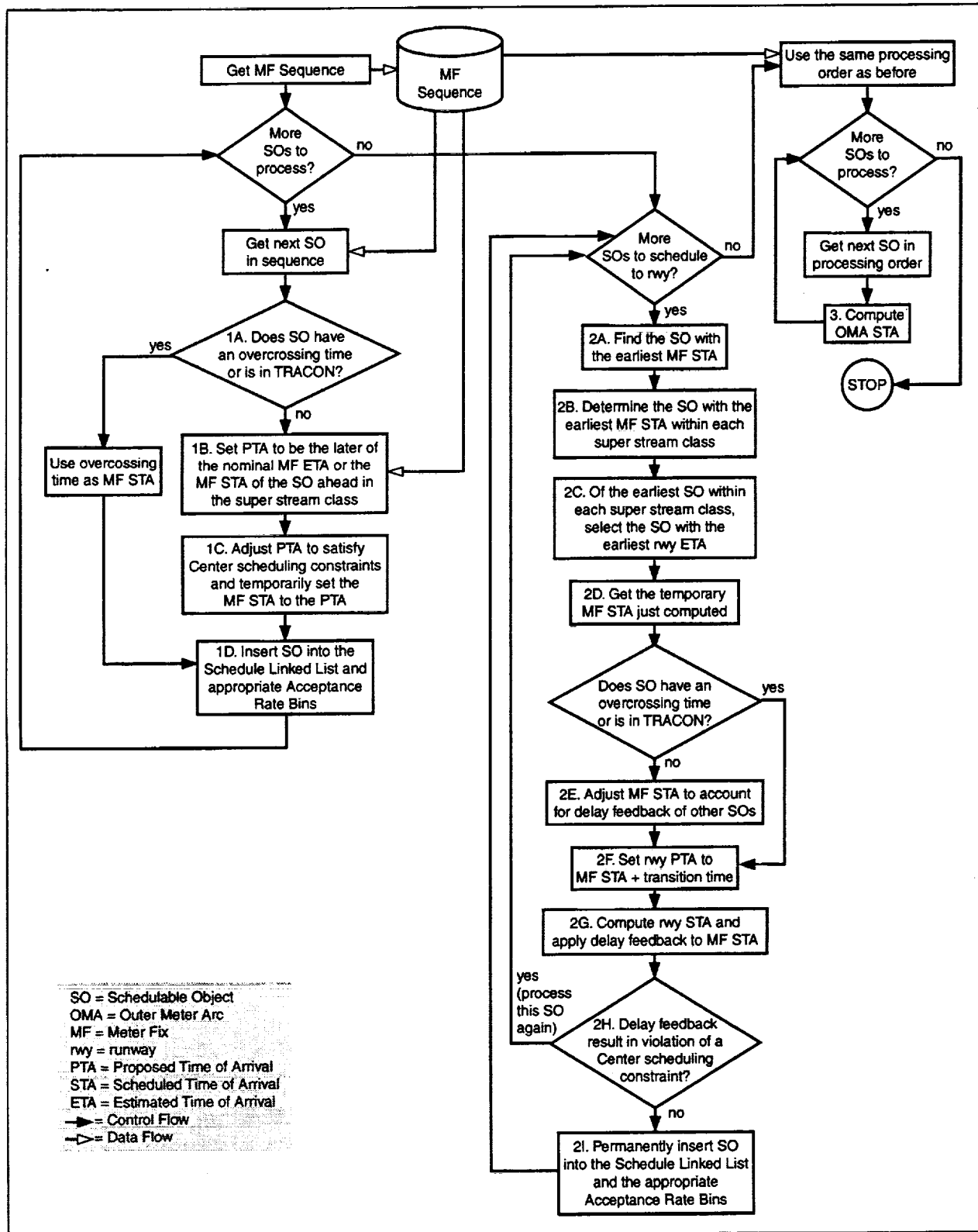▶ = Control Flow
▷ = Data Flow

Figure 38. Reschedule after Aircraft Ahead and Insert Flow Chart

Step 2B: Determine the SO with the earliest meter fix STA within each super stream class.

The time computed in the previous step is used to determine the super stream class of each SO.

Step 2C: Of the earliest SOs within each super stream class, select the one with the earliest runway ETA. The selected SO will be the next SO to consider for runway scheduling.

This mechanism generates a sequence at the runway that resembles FCFS sequencing while maintaining the sequence computed at the meter fix. By limiting the selection of the next SO to consider for runway scheduling to those with the earliest meter fix STAs, the sequence at the meter fix is maintained.

Step 2D: With the SO just selected, get the temporary meter fix STA just computed in Step 1.

Step 2E: If an overcrossing time has not been received for this SO and the SO is not within the TRACON's airspace, adjust the meter fix STA to account for delay feedback of the SO ahead in the sequence during a previous iteration of Step 2.

If the current SO has an overcrossing time or is within the TRACON's airspace, then use the overcrossing time as the meter fix STA. No adjustment of this meter fix STA is necessary since the SO has already crossed the meter fix.

If, however, this SO does not have an overcrossing time and is in the Center's airspace, then the meter fix STA may require adjustment so that it satisfies the scheduling constraints at the meter fix while remaining in its proper place in the meter fix sequence. To accomplish this, the meter fix STA of the SO ahead of the current SO in the meter fix sequence is used as a starting point. This temporary meter fix STA is then adjusted to satisfy the scheduling constraints at the meter fix. The meter fix STA of the current SO is only compared against those SOs with permanently assigned meter fix STAs. In other words, the current SO is compared against those SOs which have already had their runway STAs computed and their delay fed back to the Center. This temporary meter fix STA is compared against the meter fix STA originally computed in Step 1, and the later of the two becomes the new adjusted meter fix STA for the current SO. If the meter fix STA is modified, then adjustments are made to the Schedule Linked List and the Acceptance Rate Bins to correspond to the modified meter fix STA.

Step 2F: Set the runway PTA to the sum of the meter fix STA and the transition time from the meter fix to the final approach fix or runway threshold (see equation 1).

This is the same as Step 3C of section II.I.6.

Step 2G: Compute the SO's runway STA and apply any delay feedback to its meter fix STA.

This step is described later. For details, see section II.I.8.

Step 2H: If delay feedback caused the current SO to violate a Center scheduling constraint, then recompute the meter fix and runway STAs by going back to Step 2A.

As explained in section II.I.3, if delay feedback puts the current SO's meter fix STA in violation of a scheduling constraint, then the SO's meter fix STA must be recomputed. This is because it will be necessary to delay the meter fix STA further to avoid violating the scheduling constraint, and this new meter fix STA could make the runway STA computed in Step 2G impossible to meet. The meter fix STA plus delay feedback, after adjustment for scheduling constraints, is used as the meter fix STA when the process returns to Step 2A with the current SO.

Step 2I: If delay feedback did not cause the current SO to violate any Center scheduling constraints, then permanently insert the SO into the Schedule Link List and Acceptance Rate Bins.

At this point, the current SO has been successfully scheduled to the runway threshold, FAF, and meter fix. The current SO is permanently placed in the Schedule Link List and Acceptance Rate Bins. This means that the current SO can affect the schedules of all remaining SOs to be scheduled.

Step 3: For each SO to be scheduled using this method, compute the outer meter arc STA using equation 5.

8. Schedule to Runway

This section describes the scheduling of a single SO to the runway. The algorithm described here assumes that the meter fix STA, before delay feedback, has already been computed. It also assumes that the runway PTA has already been determined. Under VFR conditions, the scheduling constraints are applied to the FAF STAs since the aircraft are under the direction of controllers up to the FAF. Under IFR conditions, the constraints are applied to the runway threshold STAs since the controllers guide the aircraft all the way to touchdown. The process of scheduling a single SO to the runway is as follows.

Step 1: Apply the runway and airport scheduling constraints to the PTA and possibly delay the PTA.

The following runway and airport scheduling constraints are applied to the PTA.

- Airport acceptance rate
- Runway acceptance rate
- Required wake vortex separation
- Runway separation buffer
- Runway occupancy time

The resulting PTA becomes the new STA at the runway reference point (either FAF or threshold).

Step 2: Compute the STA to the other runway reference point.

If the reference point where the scheduling constraints are being applied is the FAF, then the threshold STA is computed in this step. The transition time between the FAF and the threshold is added to the STA computed in the preceding step. If the reference point where the scheduling constraints are being applied is the threshold, then the FAF STA is computed in this step. The transition time between the FAF and the threshold is subtracted from the STA computed in the preceding step.

9. Scheduling Constraints

Scheduling constraints allow the TMC to control the flow of traffic. If there are no scheduling constraints and there is no other traffic in the system, an SO will be scheduled at its nominal ETA. Realistically, however, the TMC will enter scheduling constraints that will delay an SO. That is, an SO's STA may be later than its ETA because of these scheduling constraints. The TMC will enter scheduling constraints to ensure proper spacing, favor one flow of traffic over another, compensate for adverse weather conditions, or just generally model normal air traffic control procedures.

All of the scheduling constraints are accounted for by the DP when computing STAs. Some constraints will be more restrictive than others, and these more restrictive constraints will have a greater impact on STAs than others. However, which constraint is more restrictive is not always clear. At certain times, one constraint will be more restrictive than another for certain aircraft while a different constraint will be more restrictive for other aircraft at a different time. This makes it necessary for the DP to consider all scheduling constraints when computing STAs.

The scheduling constraint's start time is the time when the constraint becomes active. SOs with ETAs that are equal to or later than the start time are affected by that constraint. Additionally, a scheduling constraint can override another scheduling constraint of the same type depending on their respective activation times. For example, suppose that a constraint limiting runway 18R's acceptance rate to 40 aircraft per hour were set to go active at time 1905Z. This is represented in figure 39

by the RWY_FC timeline tag at 05 past the hour. Further, suppose that a constraint limiting runway 18R's acceptance rate to 60 aircraft per hour were set to go active at time 1920Z. This is represented in figure 39 by the RWY_FC timeline tag at 20 past the hour. Next, suppose that UAL382 has an ETA of 1910Z while UAL365 has an ETA of 1922Z, and both aircraft are assigned to 18R. Initially, UAL382's STA will be affected by the 40 aircraft per hour constraint, while UAL365 will be affected by the 60 aircraft per hour constraint. However, if delays push UAL382's STA to 1920Z or beyond, then it will be affected by the 60 aircraft per hour constraint.
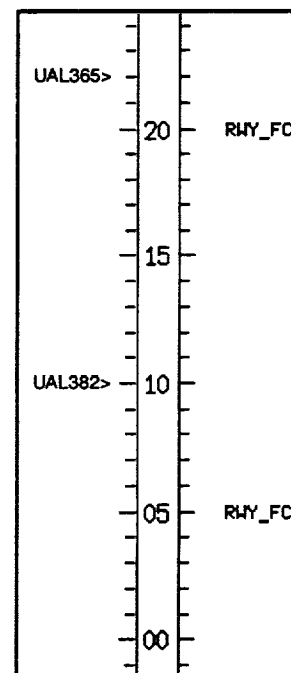
Figure 39. Runway Flow Change Example

Scheduling constraints can be divided into two airspace categories (see table 9). The Center scheduling constraints restrict the flow of traffic at the meter fixes and have a direct impact on meter fix STAs. They will also have an indirect impact on runway threshold and FAF STAs since these ultimately depend on the meter fix STAs. On the other hand, the TRACON constraints restrict the flow of traffic at the runway threshold or the FAF. Under VFR conditions, the TRACON constraints directly affect the FAF STAs. Under IFR conditions, the TRACON constraints directly affect the runway threshold STAs. Additionally, the TRACON scheduling constraints will have an effect on the meter fix STAs as a result of the feeding back of delay from the TRACON to the Center (see section II.I.3).

Table 9. Scheduling Constraints

Scheduling Constraint	Airspace Category	Scheduling Constraint Class
Meter Fix Acceptance Rate	Center	Acceptance Rate
Gate Acceptance Rate	Center	Acceptance Rate
TRACON Acceptance Rate	Center	Acceptance Rate
Super Stream Class Separation Distance (Miles-in-Trail)	Center	Separation Distance
Meter Fix Blocked Interval	Center	Blocked Interval
Airport Acceptance Rate	TRACON	Acceptance Rate
Runway Acceptance Rate	TRACON	Acceptance Rate
Runway Occupancy Time	TRACON	Occupancy Time
Wake Vortex Separation	TRACON	Separation Distance
Runway Blocked Interval	TRACON	Blocked Interval

In addition to conceptually categorizing the scheduling constraints into airspace categories, the object-oriented design of the DP divides the scheduling constraints into several classes (see table 9). Scheduling constraints are placed into a class because they share common algorithms for their application and common types of requisite data. These classes are explained in greater detail in the sections that follow.

A separate list is maintained for each type of scheduling constraint. Each list is sorted by activation time from earliest (past) to latest (future). Whenever a new Acceptance Rate constraint is added, the list is purged of constraints whose activation time precedes the current time with the exception of the constraint with the most recent activation time. The constraint with the most recent activation time is retained since it is the constraint used for the present time. The new constraint is then inserted into the list according to its activation time. However, if the new constraint has the exact same activation time as an existing constraint, then the new constraint replaces the old one. Existing constraints can be deleted from the list by specifying their activation time.

10. Separation Distance

The Separation Distance scheduling constraint restricts the horizontal distance between aircraft when they cross a reference point such as the meter fix, FAF, or runway threshold. The requisite data for this constraint are the time that this scheduling constraint is to become active and the minimum number of miles of separation required between SOs.

Because the DP's schedules are time-based, these separation distances must be converted into units of time separation. For example, for jets crossing over a meter fix, a 5 mile separation might translate into a 60 second separation. Thus, the DP would schedule one jet at least 60 seconds behind another. However, translating the separation distance to time varies from aircraft to aircraft. Although the aircraft may be flying according to some fixed airspeed, the winds aloft can affect the ground speed. Thus, one jet might cover 5 miles in 60 seconds while another might require 80 seconds to cover 5 miles because of a headwind. Therefore, the DP uses the predicted ground speed of each aircraft at the reference point to derive a reasonable translation from separation distance to time according to equation 6.

$$t_{sep} = \frac{s_{sep}}{v_g} \qquad (6)$$

s_{sep} is the required separation distance, v_g is the predicted ground speed at the reference point, and t_{sep} is the resulting time separation.

Miles-in-Trail or Super Stream Class Separation. This scheduling constraint defines the minimum allowed horizontal separation between SOs within the same super stream class and directly affects the meter fix STAs. The

separation distance, applied at the meter fix, for each super stream class is independent of the separation distance of every other super stream class. This constraint is sent to the DP along with a definition of which stream classes are placed into which super stream classes.

Wake Vortex Separation. This scheduling constraint restricts the minimum horizontal distance between SOs destined to individual or dependent runways. The airport configuration defines which runways are dependent on each other. SOs that are assigned to different but dependent runways are separated from each other as if they were assigned to the same runway.

The required separation can vary depending on the wake vortex category of the SO ahead and the SO behind. These wake vortex categories, based on weight class and engine type, are:

1. small piston,

2. small turboprop,

3. large turboprop,

4. large jet,

5. heavy jet, and

6. Boeing 757.

Note that the Boeing 757 is placed in a separate wake vortex category due to its unique wake characteristics.

The minimum required separation is specified by the user through the Wake Vortex Separation Matrix, which contains the required separation for each pair of wake vortex categories. Thus, a small piston following a large turboprop may have one required separation while a large turboprop following a small piston may have a different required separation.

The Wake Vortex Separation constraint is sent to the DP via a Runway Flow Change message. Included in this message is an optional separation buffer. The separation buffer distance is added to each value in the Wake Vortex Matrix and can be used to compensate for uncertainty in the data.

11. Acceptance Rate

The goal of the Acceptance Rate algorithm is to schedule as many SOs as possible to fully utilize, but not exceed, the Acceptance Rate. For example, suppose the only active scheduling constraint is an Airport Acceptance Rate constraint. An SO will be scheduled at its ETA unless doing so will exceed the Acceptance Rate. If the Airport Acceptance Rate is exceeded, then the SO will be

delayed to a point where its STA no longer exceeds the Airport Acceptance Rate.

At the heart of the Acceptance Rate is a data structure containing Acceptance Rate Bins. Each bin represents 30 seconds of time and contains the number of SOs scheduled within that 30 second time interval. As each SO is scheduled, the count in the bin corresponding to the SO's STA is incremented. If an SO's STA is changed as a result of delay feedback, for example, then the count in the bin corresponding to the old STA is decremented while the count in the bin corresponding to the new STA is incremented.

The algorithm begins with the current SO's PTA. This is the earliest possible time that the SO may be scheduled to cross the reference point. The 30 second bin corresponding to the PTA is determined, and a window the size of the Acceptance Rate Interval (see section II.E.3) is extended into the past from the PTA's bin. The number of SOs in each bin within this window is summed. If the total is less than the Acceptance Rate, then scheduling the current SO at the PTA would not violate the Acceptance Rate in that window. Subsequently, the window is advanced into the future by one bin. Again, the number of SOs already scheduled within the window is counted. If the number is less than the Acceptance Rate, then the window is advanced again by one bin into the future. This continues until the total count within a window equals or exceeds the acceptance rate, or the window has moved far enough into the future that it no longer contains the PTA's bin. If the total equals or exceeds the Acceptance Rate, then the PTA must be delayed. Since moving the PTA to a bin that is still within the current window would not improve the situation, the PTA is delayed to the first bin in the future just beyond the current window. Once the PTA has been delayed, the whole process of creating a sliding window and counting the number of SOs within that window is repeated. The result is a PTA that satisfies the Acceptance Rate constraint.

At the conclusion of every complete scheduling cycle, the Acceptance Rate Bins are emptied of all SOs with the exception of landed aircraft (see section II.F.5). Landed aircraft will not be scheduled in any future scheduling cycle, and they will never be re-entered into the Acceptance Rate Bins. As a result, it is necessary to maintain their presence in the Acceptance Rate Bins between scheduling cycles. However, landed aircraft are not counted against the Acceptance Rate under scheduling modes 2 through 5. Thus the Acceptance Rate Bins are completely cleared of both landed and non-landed aircraft before scheduling under these scheduling modes.

The example in figure 40 shows how the Acceptance Rate algorithm is applied. In this example, the Acceptance Rate is 20 SOs per 10 minute period. Each column of numbers represents the bins and the number of SOs which have already been scheduled into each bin.

Column (A) shows the PTA and the first window. The window is 10 minutes wide and corresponds to the 10 minute Acceptance Rate Interval. The number of SOs within the window is 15, so the window is slid up one bin at a time. Each time, the number of SOs within that window is counted.

Column (B) shows where the window is positioned when it is found that the number of SOs equals the Acceptance Rate. This means that there is no more room to schedule the current SO at its PTA. Doing so would increase the number of SOs in the window to 21 which exceeds the Acceptance Rate. The PTA is delayed to PTA'. Delaying the PTA to any bin earlier than the bin corresponding to PTA' would not help since it would still exceed the Acceptance Rate in the current window. The process of creating a window that ends at the PTA, counting SOs, and sliding the window up by one bin is repeated.
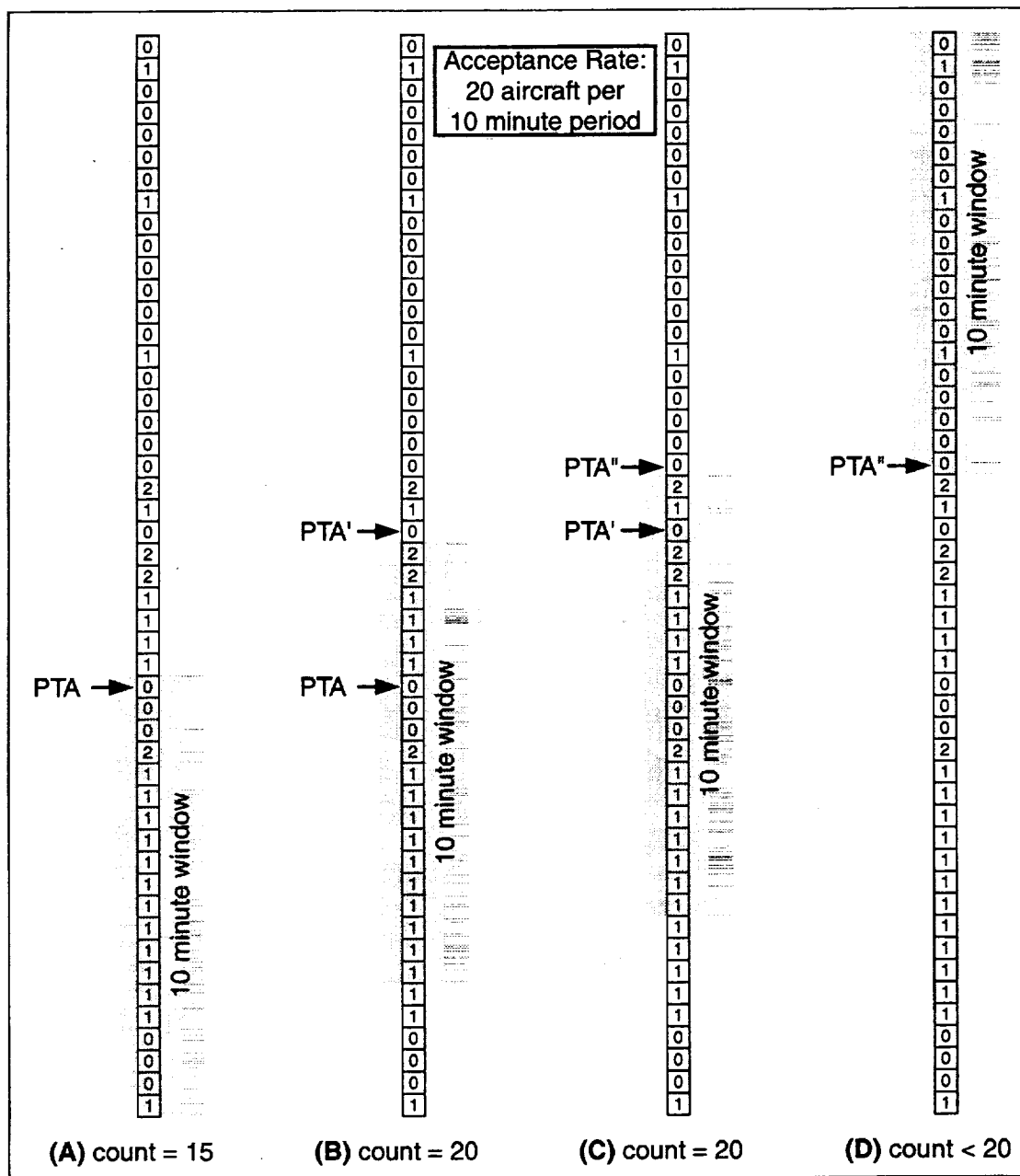
Acceptance Rate: 20 aircraft per 10 minute period

(A) count = 15   (B) count = 20   (C) count = 20   (D) count < 20

Figure 40. Acceptance Rate Example

37

Column (C) shows a window where the number of contained SOs is again 20. PTA' is then delayed to PTA" as a result. PTA" turns out to be the earliest time that does not exceed the Acceptance Rate.

Column (D) shows the last window that is checked. If there were no other scheduling constraints in the system, then the SO's STA would be set to PTA".

The algorithm translates the hourly acceptance rate, given as SOs per hour, into the number of SOs per Acceptance Rate Interval. For example, a rate of 120 SOs per hour at DFW would translate into 20 SOs per 10 minute period because the Acceptance Rate Interval is 10 minutes at DFW. In addition, the algorithm must handle the case in which the hourly acceptance rate is not evenly divisible by the number of Acceptance Rate Intervals per hour. Continuing our Dallas/Ft. Worth example, if the hourly acceptance rate were 117 SOs per hour, then this would be translated to 19 SOs per 10 minute period. However, this would result in a total acceptance rate of only 114 SOs per hour. The algorithm must try to schedule an additional 3 SOs within each hour. Currently, the algorithm schedules these remaining SOs at the beginning of each hour. Thus, in our example, 22 SOs would be scheduled to arrive during the first 10 minute period of the hour while only 19 SOs would be scheduled to arrive during each of the other 5 minute periods. As this paper is being written, feedback from the field indicates that scheduling the extra SOs to arrive during the first 10 minute period of each hour is unsatisfactory. Therefore, the handling of the remaining SOs is currently under examination, and there is currently a plan to modify the implementation so that the remaining SOs are more evenly distributed throughout the hour.

TRACON Acceptance Rate. This constraint limits the number of SOs per hour that may cross any and all meter fixes. Note that the TRACON Acceptance Rate constrains all traffic crossing the meter fixes regardless of engine type, stream class, or destination airport.

Gate Acceptance Rate. This constraint limits the number of SOs per hour that may cross any and all of the meter fixes contained within a single gate regardless of engine type, stream class, or destination airport.

Meter Fix Acceptance Rate. This constraint limits the number of SOs per hour that may cross a particular meter fix. Note that the Meter Fix Acceptance Rate constrains all traffic crossing the indicated meter fix regardless of engine type. However, different meter fix acceptance rates may be entered for the same meter fix but different destination airports. For example, an acceptance rate of 24 aircraft per hour may be entered for meter fix

BAMBE and traffic destined for DFW. At the same time, a different acceptance rate of 18 may be entered for meter fix BAMBE and traffic destined for Dallas Love Field airport (DAL).

Airport Acceptance Rate. This constraint limits the number of SOs per hour that may land at any and all runways of a particular airport.

Runway Acceptance Rate. This constraint limits the number of SOs per hour that may cross the threshold or FAF of a particular runway. This constraint is sent to the DP via the Runway Flow Change message.

12. Occupancy Time

The Occupancy Time scheduling constraint adds additional time to the required time separation translated from the Separation Distance constraint.

Runway Occupancy Time. This constraint adds additional time to the separation times translated from the Wake Vortex Separation. It is often used to account for the extra time that may be required to stop an aircraft on a slippery runway and clear it before the next aircraft lands. Alternatively, by setting all of the entries in the Wake Vortex Matrix to zero, the Runway Occupancy Time constraint can be used to separate landing SOs strictly by time.

This constraint is sent to the DP via the Runway Flow Change message. The requisite data for this constraint are the time that this scheduling constraint is to become active, the runway affected, and the number of seconds of separation to be added to the time translated from the separation distance.

13. Blocked Intervals

The Blocked Interval Scheduling Constraint prevents any SOs from being scheduled to cross a particular reference point during the specified interval of time. The Blocked Interval can be used, for example, to allow time for an airport configuration change or to avoid a severe weather cell. In addition to a start time, a Blocked Interval has an associated end time after which SOs may be scheduled.

The Blocked Interval algorithm begins with the current SO's PTA. The PTA is compared against all Blocked Intervals for the SO's assigned meter fix and runway. If the PTA is found to be between the start and end times, inclusive, of a Blocked Interval, then the PTA is delayed to the coincide with the Blocked Interval's end time.

J. Runway Allocation

Without any additional optimization, the DP, as part of TMA, has been shown to be beneficial to controllers and

TMCs. Taking the accurate ETAs generated by the RA, computing the STAs in the DP to meet the TMC's scheduling constraints, and displaying this information on the TGUI and PGUI allows TMCs to ensure a safe and smooth flow of traffic from the Center into the TRACON. It also gives the TMCs a look into the future to assist with staffing decisions.

The schedules computed by the DP can be optimized by allocating runways to SOs to reduce delay. The goal is to assign SOs to runways which reduce the delay of all SOs in the system.

1. Runway Allocation Events

From the TMC's point of view, it is undesirable to have an SO switch runways constantly even if this would continually optimize the schedule. Therefore, only certain events trigger the runway allocation process. Some of these events will cause an immediate runway allocation while others are placed in a pending list for later runway allocation. This is the same pending list as the one used by the scheduling process (see section II.I.4). When processing the events in the pending list, the DP distinguishes the scheduling events from the allocation events and takes the appropriate action.

The various runway allocation event types are listed in table 10. The runway allocation process assigns an SO to each allowable runway and generates a temporary schedule based on each runway assignment. These schedules are compared when determining which runway assignment is best. The Scheduling Modes listed in table 10 are the modes used when generating these temporary schedules, and they vary from event to event. Finally, table 10 indicates which runway allocation events are processed immediately and which are deferred for later processing.

The runway allocation event processing is summarized in figure 41. The numbers shown in figure 41 correspond to the step numbers given in this section's description.

Step 1: Determine if the current runway allocation event requires immediate runway allocation (see table 10). If this is an immediate allocation event, then go to Step 4.

Step 2: The current runway allocation event is a pending event. If the list of pending events is full or the events in the list are different from the current allocation event, then go to Step 4.

Step 3: Add this event to the list of pending events and quit.

Reaching this step means that the current allocation event is a pending event, there is room in the pending list, and this event is of the same type as those in the pending list.

This event is added to the pending list, and the runway allocation event processing is ended at this point. This event will be processed later, along with the other pending events in Step 4 in response to a different runway allocation event.

Step 4: Process the events in the list of pending allocation events (if any) and flush the list.

The SOs requiring runway allocation are collected from each pending event. Since all the pending events are of the same type, the same allocation parameters can be used for the entire set of SOs just collected. The details of the runway allocation process are explained in the next section.

Step 5: Process the current allocation event.

The details of the runway allocation process are explained in the next section.

Step 6: Perform a final reschedule.

During the runway allocation process, many partial and temporary schedules are computed. At the end of the allocation process, one final reschedule is executed to make sure each of the affected SOs has an updated schedule to its newly assigned runway.

Step 7: Prepare and send out a schedule message containing the newly computed STAs.

The scheduling message contains the STAs of the SOs. The message also indicates which runway corresponds to each STA just computed.

Step 8: Each SO's airport configuration is updated.

This is the same as Step 15 in section II.I.4.

Step 9: Flight plan amendments consisting of a change in configuration, runway, or both are sent to the other CTAS processes.

An SO may be associated with a different airport configuration than it was prior to executing the runway allocation process. Additionally, an SO may be assigned to a new runway as result of the runway allocation process. For each of these SOs, a flight plan amendment is sent notifying the rest of CTAS of the change.

Table 10. Runway Allocation Events

Runway Allocation Event Type	Scheduling Mode #	Immediate or Pending Event	Runway Allocation Mode
1. Add/delete runway blocked interval	5	Immediate	Blocked Interval Change Allocation
2. Add/delete runway flow change[a]	5	Immediate	Runway Acceptance Rate Change Allocation
3. Change in the airport configuration of an SO	5	Immediate	Allocate Within Configuration
4. TMC manually sets departure of a "proposed flight plan" aircraft in Center	9	Immediate	Allocate Within Configuration
5. SO about to become STA-Frozen (see section II.E.13)	9	Immediate	Allocate Within Configuration
6. Change in an SO's priority status	5	Immediate	Allocate Within Configuration
7. SO has been reset	9	Immediate	Allocate Within Configuration
8. Resumption of an SO that has been suspended from scheduling	9	Immediate	Allocate Within Configuration
9. A flight plan amendment is received which changes an SO's gate	9	Immediate	Allocate Within Configuration
10. A flight plan amendment is received which changes an SO's meter fix	9	Immediate	Allocate Within Configuration
11. A flight plan amendment is received which changes an SO's destination airport	9	Immediate	Allocate Within Configuration
12. Receipt of the initial set of flight plans when the DP is first brought up	2	Immediate	Allocate Within Configuration
13. Manual request to execute runway allocation for all SOs	2	Immediate	Allocate Within Configuration
14. Manual request to execute runway allocation for some SOs	3	Immediate	Allocate Within Configuration
15. Manual request to execute runway allocation for a single SO	1	Immediate	Allocate Within Configuration
16. Receipt of the first ETA for a new flight plan or newly active aircraft	9	Pending	Allocate Within Configuration

a.Only changes in the runway acceptance rates trigger runway allocation. Other changes, like occupancy time or wake vortex separation, only trigger rescheduling (see section II.I.1).
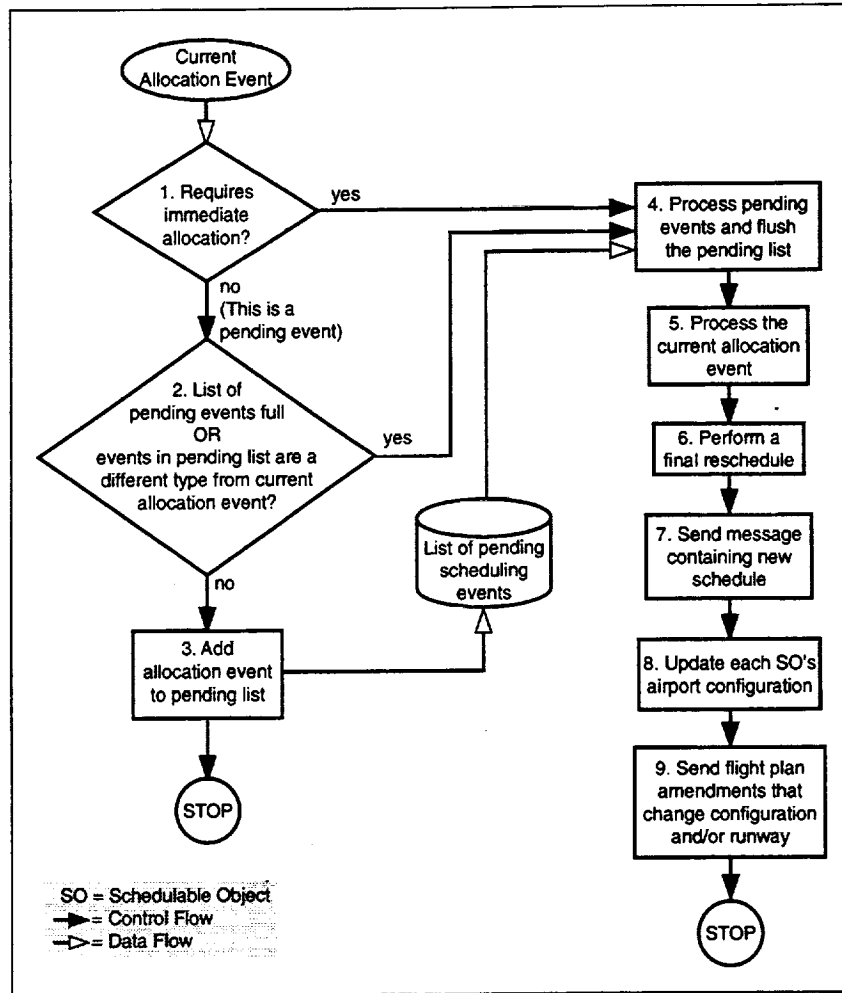
40

Figure 41. Runway Allocation Event Processing

2. Selecting SOs for Runway Allocation

Depending on the event triggering the runway allocation process, different methods are used to determine which SOs require allocation. Common to these different methods is the restriction that the following SOs do not undergo runway allocation.

- Inactive SOs with proposed flight plans
- Expired SOs
- Pop-up SOs
- Suspended SOs

Additionally, each of the selection methods described below sorts the selected SOs in order of their threshold ETAs.

Selecting a Single SO for Runway Allocation. Most runway allocation events involve a single SO (see events 3 through 11, 15, and 16 in table 10). The SO that undergoes runway allocation is specified as part of the event.

Selecting All SOs for Runway Allocation. Runway allocation event 13 in table 10 involves all SOs. In response to this event, all SOs are sent through the runway allocation process.

Selecting All SOs Following a Particular SO for Runway Allocation. Runway allocation event 14 in table 10 occurs when the user specifies that a particular SO, along with all SOs that follow it, requires runway allocation. All SOs that are destined for the same airport as the selected SO are examined. Those SOs whose runway threshold ETAs or STAs are equal to or later than the runway threshold ETA of the selected SO are put through the runway allocation process.

Selecting SOs for Runway Allocation Due to a Runway Flow Change. This method of selecting SOs for runway allocation is executed in response to a runway flow change event (event 2 in table 10) that changes the acceptance rate of one or more runways. If a runway flow change event does not change any acceptance rates,

41

then the event is treated as a rescheduling event instead (see table 3).

The SO selection process begins by determining which runways have an increasing, decreasing, or steady acceptance rate as a result of the runway flow change event. Only SOs whose runway threshold ETAs or STAs are equal to or later than the time of the runway flow change are considered for selection. In addition, SOs whose runway assignments have been locked as a result of manual runway assignments are not selected. From the remaining SOs, the selection process determines which SOs require runway allocation depending on which of the following conditions is true.

Condition 1: Exactly one runway has an increasing acceptance rate, and the acceptance rates of the other runways are steady or decreasing.

SOs from runways with decreasing acceptance rates and Non-STA-Frozen SOs on steady acceptance rate runways are selected.

Condition 2: More than one runway has an increasing acceptance rate.

SOs from runways with decreasing or increasing acceptance rates and Non-STA-Frozen SOs on steady acceptance rate runways are selected.

Condition 3: One or more runways has a decreasing acceptance rate, and no runways have an increasing acceptance rate.

SOs from runways with decreasing acceptance rates are selected.

Selecting SOs for Runway Allocation Due to the Addition or Deletion of a Runway Blocked Interval. This method of selecting SOs for runway allocation is executed in response to the addition or deletion of a runway blocked interval (event 1 in table 10). Only SOs whose runway threshold ETAs or STAs are between the start and stop time of the blocked interval (inclusive) are considered for selection. From this group of SOs, this method selects SOs for runway allocation in a manner similar to that used when a runway flow change is added or deleted (see above). In this case, the addition of a blocked interval to a runway is similar to reducing the acceptance rate of that runway, and the deletion of a blocked interval is similar to increasing the acceptance rate.

3. Runway Allocation Process

Once a runway allocation event has been received and the SOs that require allocation have been selected, the actual runway allocation process is executed on the selected SOs. The SOs are processed in order of

increasing runway threshold ETAs. Placing the SOs in this order is the responsibility of the SO selection methods described in the previous section.

The runway allocation process is summarized in figure 42. The numbers shown in figure 42 correspond to the step numbers given in the this section's description.
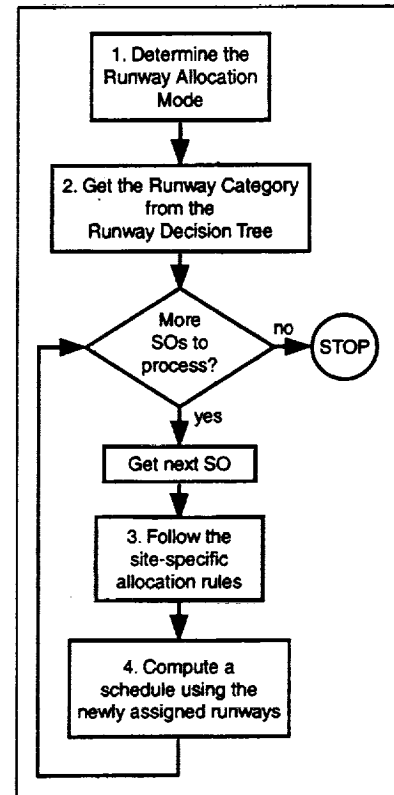

Figure 42. Runway Allocation Process

Step 1: Determine the Runway Allocation Mode.

Many of the runway allocation events trigger similar runway allocation methods. Those events with similar allocation methods are grouped together into Runway Allocation Modes. There are currently three runway allocation modes, and the mapping from runway allocation events to runway allocation modes is shown in table 10. The Runway Allocation Mode derived from the runway allocation event is then used in the next step.

Step 2: Get the Runway Category from the Runway Decision Tree.

Decision trees are used throughout CTAS as a way of encoding site-dependent rules. Similar in functionality to nested case statements in C, a decision tree identifies a category based on a set of criteria. Because of its generic nature, decision trees can be applied in many different ways and can vary from site to site. The category that

results from the traversal of a decision tree can then be used to identify which set of rules is to be followed. Thus, decision trees provide a mechanism to execute a set of rules specifically designed for a certain set of criteria.

Runway allocation is one of these CTAS processes that utilize a decision tree. The runway allocation decision tree is contained in a site-dependent data file called *dp_runway_decision_tree*. The decision tree is traversed for each SO to determine its runway category. The criteria used by the runway decision tree are listed below.

- SO's destination airport
- Airport configuration for the SO
- SO's meter fix
- SO's engine type

An excerpt from the *dp_runway_decision_tree* file is shown in listing 1. Suppose that there has been a change to the SOUTH_4_VFR airport configuration, and United 242 requires runway allocation. Further, suppose that United 242 is a Boeing 747 destined for DFW and assigned to the BAMBE meter fix. The first line in the file asks for which airport United 242 is destined. In this case, the destination is DFW. Next, the decision tree asks to which meter fix has United 242 been assigned. United 242 has been assigned to BAMBE, so the runway allocation mode must be determined. The event triggering runway allocation is an airport configuration

change, and according to table 10, the runway allocation mode is ALLOCATE_WITHIN_CONFIGURATION. Next, the airport configuration is examined. Since SOUTH_4_VFR is one of the configurations listed, United 242's engine type is examined. Since United 242 is neither a turbo prop nor a piston aircraft, the default value is used. The resulting runway allocation category is DFW_BA_S_4_13R_DEF, and this is used in Step 3.

Step 3: For each SO, follow the site-specific allocation rules.

Associated with each runway category is a set of rules for determining the best runway for a particular SO. These rules are contained in the site-dependent data file called *dp_runway_category_definitions*. If, after following these rules, it is determined that the best runway for a particular SO is different from the runway previously assigned, then a flight plan amendment is sent to the other CTAS processes informing them of the change in runway assignment.

An excerpt from the *dp_runway_category_definitions* file is shown in listing 2. Continuing with the United 242 example, this excerpt shows the runway category, called DFW_BA_S_4_13R_DEF, determined from the previous step. The rules in this category are processed from top to bottom. Each of these rules is explained below.

```
criteria WHICH_AIRPORT {
    value DFW
        criteria WHICH_METER_FIX {
            value BAMBE
                criteria WHICH_RWY_ALLOC_MODE {
                    value ALLOCATE_WITHIN_CONFIGURATION
                        criteria WHICH_CONFIGURATION {
                            value SOUTH_4_VFR
                            value SOUTH_4_IFR
                            value SOUTH_4_VFR_BA_GG_REROUTE
                            value SOUTH_4_IFR_BA_GG_REROUTE
                                criteria WHICH_ENGINE{
                                    value TURBO_PROP
                                    value PISTON
                                        category DFW_BA_S_4_13R_TURBO_DEF
                                    value DEFAULT
                                        category DFW_BA_S_4_13R_DEF
                }
```

Listing 1. *dp_runway_decision_tree* Data File Excerpt

```
category DFW_BA_S_4_13R_DEF
        assign_runway       13R
            check_criteria  DEFAULT_RUNWAY                  {}
        assign_runway       18R
            check_criteria  USE_RUNWAY_IF_DELAY_REDUCED     {0.0}
        assign_runway       17C
            check_criteria  USE_RUNWAY_IF_DELAY_REDUCED     {1.0}
        assign_runway       17L
            from_runway     17C
            check_criteria  USE_RUNWAY_IF_DELAY_REDUCED     {1.0}
```

Listing 2. *dp_runway_category_definitions* Data File Excerpt

assign_runway. This specifies which runway to assign to the SO if the criteria listed are met. The name of the runway will follow the keywords assign_runway, or the words PREVIOUSLY_ALLOCATED_RWY will appear in place of the runway name. In the latter case, the runway to which the SO had been previously assigned is the one that is examined.

from_runway R. This criterion is true if the best runway chosen so far is R. For example, in the last assign_runway block in listing 2, in order to accept 17L as the best runway, the best runway chosen so far must be 17C plus the result of check_criteria which is explained below.

check_criteria. This specifies a criterion to be met for this runway assignment to be the best runway chosen so far. It is followed by a rule and possibly a value contained in curly braces. These rules are as follows.

> **DEFAULT_RUNWAY {}.** This criterion is always true. When this criterion appears in an assign_runway block no other criterion can be contained within that block. It is used to specify the runway to assign to the aircraft if all of the other possible assignments are less desirable. The SO is temporarily assigned to this runway, and a temporary schedule is computed. The runway threshold STAs of all SOs are added together, and this sum is known as the System Schedule Time.

For example, in listing 2, 13R is initially considered the best runway. This may change as the subsequent assign_runway blocks are processed.

> **USE_RUNWAY_IF_DELAY_REDUCED {D}.** The SO is temporarily assigned to the runway specified on the immediately preceding assign_runway line. A temporary schedule is computed along with the associated System Schedule Time. If this System Schedule Time is less

than the System Schedule Time of the best runway assignment seen so far by D minutes, then this criterion is true. If the other criteria within the assign_runway block hold up, then this runway becomes the best runway seen so far.

For example, in listing 2, 13R is initially the best runway. In the second assign_runway block, the SO is assigned to 18R and a schedule is computed. If the System Schedule Time associated with assigning the SO to 18R is better than the System Schedule Time associated with assigning the SO to 13R by more than 0.0 minutes, then 18R becomes the best runway so far.

The number of minutes D can be used to prevent an SO from jumping back and forth between runways just because the System Schedule Time is reduced by a couple of seconds.

> **RUNWAY_ACCEPTANCE_RATE_CHANGE {}.** This criterion is true if either of the following conditions is true.
>
> - The SO's original runway has an increasing or steady acceptance rate and the runway specified in assign_runway has an increasing acceptance rate.
> - The SO's original runway has a decreasing acceptance rate and the runway specified in assign_runway has an increasing or steady acceptance rate.

This criterion steers the runway selection to runways that have improving acceptance rates when compared with the SO's original runway. For example, if one runway has an increasing acceptance rate, then SOs on other runways need only compare their original runways against the runway with the increasing acceptance rate. As another example, if the SO is originally on a runway with a decreasing acceptance rate, then all runways with increasing or

steady acceptance rates are candidates for assignment.

ADD_OR_REMOVE_BLOCKED_INTERVAL {}.
Under the current implementation, this criterion is treated the same as RUNWAY_ACCEPTANCE_RATE_CHANGE with the idea that adding a blocked interval is the same as decreasing a runway's acceptance rate, and deleting a blocked interval is the same as increasing a runway's acceptance rate.

Continuing the example of United 242, the first line in listing 2 tells the runway allocator to first assign United 242 to runway 13R. The resulting schedule is computed, and let's say the associated System Schedule Time is 45000.0 minutes. Next, United 242 is temporarily assigned to runway 18R. Another schedule is computed, and the associated System Schedule Time is 44996.0 minutes. Since the parameter to the USE_RUNWAY_IF_DELAY_REDUCED criterion is 0.0, runway 18R will become the best runway examined so far if the new System Schedule Time is less than the System Schedule Time computed for runway 13R. Indeed, the System Schedule Time for 18R is less than that for 13R, so 18R becomes the best runway examined so far. Next, United 242 is temporarily assigned to runway 17C, and the resulting System Schedule Time is 44995.5 minutes. Although the new System Schedule Time is less than that computed for runway 18R, the difference does not exceed the USE_RUNWAY_IF_DELAY_REDUCED parameter of 1.0. Therefore, 18R remains the best runway examined so far. Finally, the last assign_runway block is processed. The criterion from_runway 17C means that 17L is only considered if the best runway so far is 17C. This is not the case, so the runway allocator does not need to consider runway 17L. The final result is that United 242 is assigned to runway 18R. If 18R is different from the runway that United 242 was assigned to before the execution of the runway allocation process, then a flight plan amendment is sent to the other CTAS processes informing them of the change in runway assignment.

Step 4: Compute a schedule using the newly assigned runways.

Once all of the SOs involved in runway allocation have been assigned to runways, a final schedule is computed (see section II.I.4). Subsequently, a schedule message containing the new STAs is sent to the rest of CTAS.

K. Miles-in-Trail Advisor

The Miles-in-Trail Advisor (MINTA) functionality in the DP analyzes the future traffic flow and computes the super stream class separations (Miles-in-Trail) to meet a TRACON acceptance rate specified by the TMC. Less busy super stream classes will be given larger separations in order to relieve the pressure from the busier super stream classes. This functionality is still undergoing research and development, but the current functionality is described below.

The TMC, through the TGUI, sends a Miles-in-Trail request which includes the following information.

Start Time and Stop Time. The start time and stop time specify the period for which the advisory is to be used. The MINTA functionality will analyze aircraft only from this period when formulating a response to the TMC's request.

Desired TRACON Acceptance Rate (AR_{TRACON}). The MINTA functionality will compute the separation distances for each super stream class so that the desired TRACON acceptance rate is fully utilized.

Super Stream Class Definitions. These definitions specify which stream classes are grouped into which super stream classes for the time period specified above. This is similar to specifying the super stream class separation distance scheduling constraint (see section II.I.10).

Manual Separation Distances. The TMC may specify the desired separation distance for zero or more super stream classes. MINTA will compute the separation distances for the remaining super stream classes such that the TRACON acceptance rate is fully utilized.

Once the parameters have been specified by the TMC, MINTA first processes the super stream classes for which the TMC has specified the separation distances. These super stream classes are known as Specified Super Stream Classes. The average ground speed of qualified aircraft (not blocked slots) in each of these Specified Super Stream Classes is computed. To be qualified, an aircraft must have a non-zero ground speed[5] at the meter fix. Additionally, if the aircraft is STA-Frozen, then its meter fix STA must be between the start and stop times specified by the TMC. If the aircraft is not STA-Frozen, then its nominal meter fix ETA must be between the start and stop times specified by the TMC. For each Specified Super Stream Class, i, the average ground speed, $\bar{s}_i$, is used to convert the specified separation distance, d_i, to an

---

[5] A ground speed of zero is an indication that the Route Analysis (RA) and Trajectory Synthesizer (TS) programs failed to compute a ground speed for that aircraft.

acceptance rate, AR_i, according to the following equation.

$$AR_i = \frac{\bar{s_i}}{d_i} \qquad (7)$$

Let P be the number of Specified Super Stream Classes. The acceptance rates for the P Specified Super Stream Classes are subtracted from the desired TRACON acceptance rate, AR_{TRACON}. The result, AR_{req}, is the total TRACON acceptance rate to be apportioned to the super stream classes for which the TMC has not specified a separation distance. However, for practical reasons, the minimum that AR_{req} can be is 10% of the desired TRACON acceptance rate AR_{TRACON} (see equation 10).

$$\alpha = AR_{TRACON} - \sum_i^P AR_i \qquad (8)$$

$$\beta = (0.1)AR_{TRACON} \qquad (9)$$

$$AR_{req} = MAX(\alpha, \beta) \qquad (10)$$

Next, the remaining super stream classes, which did not have a separation distance specified for them, are processed. These super stream classes are known as Requested Super Stream Classes. The remaining acceptance rate, AR_{req}, is divided among the Requested Super Stream Classes in proportion to the number of aircraft in each super stream class. Because the minimum that AR_{req} can be is 10% of the desired TRACON acceptance rate, AR_{TRACON}, it is possible that the separation distances computed by MINTA may result in a TRACON Acceptance Rate that exceeds the desired TRACON Acceptance Rate, AR_{TRACON}. The number of qualifying aircraft is counted for each super stream class. Simultaneously, the average ground speed for each super stream class is computed for use during a later step when the acceptance rate is converted to a separation distance. For example, suppose there are three super stream classes. Further suppose that the first super stream class has 10 aircraft within the specified time period, and the other two super stream classes have 5 aircraft each. Since the first super stream classes accounts for 50% of the total traffic, 50% of AR_{req} is apportioned to the first super stream class. So, if AR_{req} is 60 aircraft per hour, then the first super stream class is given an acceptance rate, AR_j, of 30. The other two super stream classes are each given a rate of 15.

Once the acceptance rates have been apportioned to the Requested Super Stream Classes, the separation distance for each of the requested super stream classes is computed. For each Requested Super Stream Class, j, the average ground speed, $\bar{s_j}$, is used to convert the

acceptance rate, AR_j, to the separation distance d_j, according to the following equation.

$$d_j = \frac{\bar{s_j}}{AR_{req}} \qquad (11)$$

Finally, a response message is created. This message contains separation distances for each of the super stream classes. Note that if a Requested Super Stream Class has no qualifying aircraft, then the average ground speed and, hence, the separation distance cannot be computed for that Requested Super Stream Class. In this case, an exceptional value is stored in the message. The recipient recognizes this exceptional value as an indication that a separation distance was not successfully computed for a particular super stream class. The message also contains a flag indicating if MINTA was successful in computing the separation distances given the inputs and the data available. In addition, the message contains an indication of whether the desired TRACON acceptance rate, AR_{TRACON}, is estimated to be exceeded if the recommended separation distances are followed. This is due to the provision in the algorithm that guarantees that a minimum of 10% of the desired TRACON acceptance rate (see equation 9 and equation 10) is apportioned to each super stream class.

L. ETA Hovering

ETA Hovering is a mechanism that the DP uses to compensate for any inaccuracies in the coordination fix time contained in a flight plan. Before CTAS receives active tracks for an aircraft, the RA will compute the aircraft's ETA based on information contained in the aircraft's flight plan. Sometimes, however, an aircraft will become active at a different time than indicated by the flight plan. If left alone, the ETA based on the inactive aircraft's flight plan will move inside the STA Freeze Horizon. The STA, which is based on this ETA, may be frozen at a time as early as the ETA. If the aircraft becomes active much later than indicated by the flight plan, then the new active aircraft ETA computed by RA will be much later than the ETA computed for the flight plan. This will make it impossible for the aircraft to meet the STA which was computed while the aircraft was inactive.

The ETA hovering mechanism hovers the ETA of such an inactive aircraft outside the STA Freeze Horizon. Periodically, the DP checks all of the aircraft and hovers the ETAs of those aircraft which are eligible. If an eligible aircraft is found to be within the ETA Hover Horizon, then the ETA Hover Amount is added to the aircraft's ETA. The ETA Hover Horizon is currently set to be 180 seconds outside the STA Freeze Horizon, and

this value may be changed by the user as appropriate for the deployment site. The additions to the ETA are maintained in the DP. The other CTAS processes are not aware of the change in ETA except via the hovering's effect on the aircraft's STA. It's possible that an eligible aircraft may be hovered several times to keep it outside of the STA Freeze Horizon until it goes active.

ETA hovering is applied only to aircraft. Blocked slots never have their ETAs hovered. Moreover, only aircraft which satisfy all of the following criteria are eligible for ETA hovering:

- The aircraft must be inactive. That is, no tracks have been received, and the aircraft's ETA is computed using information contained in its flight plan.

- The aircraft cannot be STA-Frozen.

- The aircraft has not landed (see section II.F.5).

- The aircraft is not a pop-up (see section II.F.3).

- The aircraft has not been manually departed (see section II.F.1).

- The aircraft is not a proposed flight plan.

- The aircraft is not awaiting a new ETA as a result of a flight plan amendment. It is possible that the new ETA will make an aircraft ineligible for ETA hovering, so the DP waits until the ETA is received before hovering the aircraft.

- The aircraft is not a departed flight plan.

- The aircraft's flight time to the meter fix is not less than the ETA Hover Horizon.

M. Broadcast

The Communications Manager (CM), in response to certain events, will prevent the sending of STAs to the controllers' Planview Displays (PVDs) and PGUIs via a mechanism known as Broadcast Blocking. The events which trigger Broadcast Blocking are those which have a direct effect on the STAs of STA-Frozen SOs. Since only STAs of STA-Frozen aircraft are displayed on the controllers' PVDs, these are the type of changes that will be noticed by the sector controllers. Such changes are referred to as an aircraft list "ripple." Broadcast Blocking allows the TMC to make multiple adjustments to the traffic flow and make sure all of the sector controllers are ready for the inevitable "ripple." The DP is not even aware that schedules are blocked. The DP continues to generate schedules and these schedules are updated on the TMC's TGUI and PGUI. The responsibility of blocking the schedules is left to the CM through which all message traffic flows. To turn Broadcast Blocking off, the TMC issues a Broadcast All command from the TGUI. When the DP receives the request to Broadcast All, it prepares a schedule message containing the STAs

of all aircraft and sends it to the CM. The CM will turn the Broadcast Blocking off and forward the DP's schedule message to all PGUIs, TGUIs, and the two-way interface.

In addition to the Broadcast All command, the TMC may request that the schedule for a particular aircraft be broadcast using a Broadcast <acid> command. When the DP receives such a request, it prepares a schedule message containing the scheduling information for the requested aircraft. The CM will forward this schedule message to all PGUIs, TGUIs, and the two-way interface. However, CM will not turn the Broadcast Blocking off. Turning the Broadcast Blocking off requires that the TMC issue the Broadcast All command.

N. Design Methodology

A team of software engineers, aerospace researchers, and air traffic control experts was formed to establish the requirements for the DP. The requirements reflected the lessons learned from previous scheduler implementations, the experience gained in the field, and the possible direction of future research.

Once the requirements were established, a team of software engineers applied the techniques of Object-Oriented Analysis (OOA) and Object-Oriented Design (OOD) as described by Rumbaugh et al. [18]. OOA and OOD techniques were selected because of the ease with which the implementation task could be divided among the programming resources as well as the ease with which the design could be maintained. OOA and OOD also resulted in a design that was flexible enough to meet the dynamic requirements of the air traffic control researchers.

The resulting design was implemented in ANSI C. Although C is not an Object-Oriented Programming (OOP) language, the software engineering team established a set of programming guidelines that resulted in C code that resembled an object-oriented implementation. The software engineering team as well as future software developers must maintain the discipline to follow these guidelines in order to maintain an object-oriented program since the C language offers very few tools to enforce object-oriented programming.

III. CONCLUDING REMARKS

The DP computes the aircraft sequence, STAs, and runway assignments to ensure an orderly, efficient, and conflict-free flow of traffic into the terminal area as part of the TMA tool of CTAS. The DP sequences the aircraft so that they arrive in an FCFS order at the meter fix unless the TMC overrides this order via manually entered sequence constraints. Also, the STAs computed

by the DP meet all of the spatial and flow constraints entered by the TMC. These constraints reflect the current and anticipated runway capacities, traffic densities and distributions, airport configurations, and weather conditions. The DP assigns aircraft to the active runways to optimize the schedule while taking into account any operational runway assignment procedures. The above results are updated in response to changing events and TMC inputs at a rate comparable to the live radar update rate.

OOA and OOD methods were used in the design of the DP. This has made it relatively easy to modify the DP in response to feedback from the field. The DP is the first component of CTAS to use object-oriented techniques in its design, and its success has led to a wider use of object-oriented techniques in the design of other CTAS processes.

The DP continues to evolve as new research is conducted. Research is currently being conducted to improve the Miles-in-Trail Advisor [19], improve the fairness in the distribution of delay between meter fixes, and basing the STA Freeze Horizon on the location of the aircraft as opposed to its ETA. Future work includes making the modifications necessary to fully integrate TMA with FAST, and adding a method of detecting when aircraft have been placed into holding, and its effect on the STAs of other aircraft.

The DP has been installed at both Denver Center [20] and Ft. Worth Center for testing and evaluation [4]. Feedback from the field has been both positive and useful, and a number of changes have been implemented in response to requests from the field.

Currently, the DP is in daily use as a flow-visualization tool at Atlanta, Denver, Los Angeles, and Miami Centers. In addition, the DP is in daily use at Ft. Worth Center as the primary arrival planning tool. In the near future, the DP will also be the primary arrival planning tool for Altanta, Denver, Los Angeles, and Miami Centers.

IV. REFERENCES

1. Denery, D. G.; and Erzberger, H.: The Center-TRACON Automation System: Simulation and Field Testing. In: Proceedings of the advanced Workshop on ATM (ATM 95), Capri, Italy, Oct. 2-6, 1995. Also published as NASA TM-110366, Aug., 1995.

2. Erzberger, H.:, Statement of Dr. Heinz Erzberger of NASA Concerning the Center-TRACON Automation System (CTAS). Pesented at the Meeting of the Congressional Capacity Technology Subcommittee, FAA Research and Development Advisory Committee, Washington, DC, July 12, 1994.

3. Sanford, B.D.; Harwood, K.; Nowlin, S.; Bergeron, H.; Heinrichs, H.; Wells, G.; and Hart, M.: Center/TRACON Automation System: Development and Evaluation in the Field. In: 38th Annual Air Traffic Control Association Conference Proceedings, October, 1993.

4. Swenson, H. N.; Hoang, T.; Engelland, S.; Vincent, D.; Sanders, T.; Sanford, B.; and Heere, K.: Design and Operational Evaluation of the Traffic Management Advisor at the Ft. Worth Air Route Traffic Control Center. In: Proceedings of the 1st US/Europe ATM R&D Seminar, Saclay, France, June, 1997.

5. Erzberger, H.: Design Principles and Algorithms for Automated Air Traffic Management. In: AGARD Lecture Series 200, Madrid, Spain, Paris, France, and Moffett Field, California, USA, November, 1995.

6. Nedell, W.; and Erzberger, H.: The Traffic Management Advisor. Presented at the 1990 American Control Conference, San Diego, CA, May, 1990.

7. Green, S. M.; Vivona, R.; and Sanford, B.: Descent Advisor Preliminary Field Test. AIAA-95-3368, August, 1995.

8. Davis, T. J.; Isaacson, D. R.; Robinson III, J. E.; den Braven, W.; Lee, K. K.; and Sanford, B.: Operational Test Results of the Passive Final Approach Spacing Tool. Presented at the IFAC 8th Symposium on Transportation Systems, Chania, Greece, June, 1997.

9. Krzeczowski, K. J.; Davis, T. J.; Erzberger, H.; Lev-Ram, I.; and Bergh, C. P.: Knowledge-Based Scheduling of Arrival Aircraft in the Terminal Area. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, Paper No. AIAA-95-3366, Baltimore, MD, August, 1995.

10. Davis, T.J.; Krzeczowski, K. J.; and Bergh, C. P.: The Final Approach Spacing Tool. Presented at the IFAC Thirteenth Symposium on Automatic Control in Aerospace, Palo Alto, CA, September, 1994.

11. Paielli, R. A.; and Erzberger, H.: Conflict Probability Estimation for Free Flight. In: Journal of Guidance, Control, and Dynamics, Vol. 20, No. 3, May-June, 1997. Also published as NASA TM-110411.

12. Erzberger, H.; Paielli, R. A.; Isaacson, D. R.; and Eshow, M. M.: Conflict Detection and Resolution In the Presence of Prediction Error. Presented at the 1st USA/Europe Air Traffic Management R&D Seminar, Paris, France, June, 1997.

13. Sokkappa, B. G.; and Steinbacher, J. G.: Performance of En Route Metering Methods. MITRE Technical Report MTR-7575, June, 1977.

14. Petre, E.: Time Based Air Traffic Control in an Extended Terminal Area - A Survey of Such Systems. Doc. 912009, Eurocontrol, Bruxelles, Belgium, June, 1991.

15. Brinton, C. R.: An Implicit Enumeration Algorithm for Arrival Aircraft Scheduling. Presented at the Eleventh Digital Avionics Systems Conference, Seattle, WA, October, 1992.

16. Slattery, R.; and Zhao, Y.: Trajectory Synthesis for Air Traffic Automation. In: Journal of Guidance, Control, and Dynamics, Vol. 20, No. 2, pp. 232-238, March/April, 1997.

17. Slattery, R.A.: Terminal Area Trajectory Synthesis for Air Traffic Control Automation. Submitted to the American Control Conference, June, 1995.

18. Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F.; and Lorensen, W.: Object-Oriented Modeling and Design. Prentice Hall, Englewood Cliffs, NJ, 1991.

19. Synnestvedt, R. G.; Swenson, H. N.; and Erzberger, H.: Scheduling Logic for Miles-In-Trail Traffic Management. NASA Technical Memorandum TM-4700, September, 1995.

20. Harwood, K.; and Sanford, B.: Denver TMA Assessment. NASA Contractor Report 4554, October, 1993.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE April 2000	3. REPORT TYPE AND DATES COVERED Technical Memorandum

4. TITLE AND SUBTITLE

The Dynamic Planner: The Sequencer, Scheduler, and Runway Allocator for Air Traffic Control Automation

5. FUNDING NUMBERS

538-18-24

6. AUTHOR(S)

Gregory L. Wong

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Ames Research Center
Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER

A-00V0008

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA/TM–2000-209586

11. SUPPLEMENTARY NOTES

Point of Contact: Gregory L. Wong, Ames Research Center, MS 210-8, Moffett Field, CA 94035-1000
(650) 604-1439

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified — Unlimited
Subject Category 03, 61
Availability: NASA CASI (301) 621-0390

Distribution: Standard

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Dynamic Planner (DP) has been designed, implemented, and integrated into the Center-TRACON Automation System (CTAS) to assist Traffic Management Coordinators (TMCs), in real time, with the task of planning and scheduling arrival traffic approximately 35 to 200 nautical miles from the destination airport. The TMC may input to the DP a series of current and future scheduling constraints that reflect the operation and environmental conditions of the airspace. Under these constraints, the DP uses flight plans, track updates, and Estimated Time of Arrival (ETA) predictions to calculate optimal runway assignments and arrival schedules that help ensure an orderly, efficient, and conflict-free flow of traffic into the terminal area. These runway assignments and schedules can be shown directly to controllers or they can be used by other CTAS tools to generate advisories to the controllers. Additionally, the TMC and controllers may override the decisions made by the DP for tactical considerations. The DP will adapt to computations to accommodate these manual inputs.

14. SUBJECT TERMS

Air traffic control, Center -TRACON Automation System, Scheduling, Traffic Management advisor

15. NUMBER OF PAGES

57

16. PRICE CODE

A04

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102