

FLIGHT TRANSPORTATION LABORATORY

REPORT R86-6

INTERACTIVE DYNAMIC AIRCRAFT
SCHEDULING AND FLEET ROUTING WITH THE
OUT-OF-KILTER ALGORITHM

JAN VAN COTTHEM

**INTERACTIVE DYNAMIC AIRCRAFT SCHEDULING AND
FLEET ROUTING WITH THE OUT-OF-KILTER ALGORITHM**

by

JAN VAN COTTHEM

ABSTRACT

A decision support system is introduced that automates dynamic aircraft scheduling and fleet routing. Interactive graphics-based schedule construction and modification tools automate the dynamic scheduling of aircraft of a single-type aircraft fleet and the out-of-kilter algorithm is employed to automate their routing. With the out-of-kilter algorithm, either the minimum fleet size required to serve a complete schedule or the routes that aircraft of a fleet of fixed size should serve in order to maximize fleet income can be determined. Since many scheduling scenarios can be easily constructed and evaluated, fleet routing solutions that achieve planning goals, while satisfying organizational constraints, can be quickly obtained. As a result, significant improvements in aircraft scheduling and fleet routing productivity and quality are possible.

ACKNOWLEDGEMENTS

Programming the software for the decision support system presented in this thesis was largely a painstaking and solitary activity. Nonetheless, I would like to thank the people who contributed in various ways to the undertaking: Professor Simpson, who proposed additions to the software and corrected my faulty reasoning, D.F.X Mathaisel, Research Engineer, who demonstrated an intermediate version of the program to an airline, Boon Chai Lee, graduate student, who in the course of many discussions on the topic, made several helpful suggestions, and Dionyssios Trivizas, graduate student, who introduced me to the L^AT_EX word processor.

LIST OF FIGURES

Figure 1 - The kilter diagram of arc ij.	41
Figure 2 - Acceptable kilter states for forward and reverse arcs to be included in the flow-augmenting path.	42
Figure 3 - Kilter states of arcs linking labelled and unlabelled nodes of which the node multipliers will be changed.	43
Figure 4 - Daily schedule plan schedule chart.	44
Figure 5 - Multiple-day schedule plan schedule chart.	45
Figure 6 - Network arc parameters by schedule plan and optimization option.	46
Figure 7 - Mathematical formulations by schedule plan and optimization option.	47
Figure 8 - Iteration scenario when specified fleet size is less than maximum fleet size.	48
Figure 9 - Segment number input.	49
Figure 10 - Origin station code name input.	50
Figure 11 - Destination station code name input.	51
Figure 12 - Flight segment value input.	52
Figure 13 - Flight segment duration input.	53
Figure 14 - The flight segment rectangle is moved to the appropriate position.	54
Figure 15 - Flight segment 101 has been scheduled.	55
Figure 16 - Flight segment characteristics display.	56
Figure 17 - Flight segment 101's value is increased.	57
Figure 18 - A schedule plan consisting of four flight segments.	58

Figure 19 - Flight segment 104 is about to be selected for rescheduling.	59
Figure 20 - Flight segment 104 is being rescheduled.	60
Figure 21 - Flight segment 104 is about to be deleted.	61
Figure 22 - Flight segment 104 has been deleted.	62
Figure 23 - A new time interval is selected	63
Figure 24 - The visible portion of the schedule chart has been changed.	64
Figure 25 - The schedule plan of Figure 23 is tabulated.	65
Figure 26 - Station display for stations AAA and BBB.	66
Figure 27 - Flight segment 103 will be deleted.	67
Figure 28 - Flight segment 103 has been deleted.	68
Figure 29 - Flight segment 107 is being added to the schedule plan.	69
Figure 30 - The value of flight segment 107 is 350.	70
Figure 31 - The duration of the flight segment is 2.5 hrs.	71
Figure 32 - Flight segment 107's departure time is selected.	72
Figure 33 - Flight segment 107 has been rescheduled.	73
Figure 34 - The "time fields" for the selection of arrival or departure times.	74
Figure 35 - Flight segment 101 is selected for rescheduling.	75
Figure 36 - The arrow is moved to the new departure time.	76
Figure 37 - Flight segment 101 has been rescheduled.	77
Figure 38 - The time axis of either AAA or BBB can be translated.	78

Figure 39 - BBB's visible time axis interval will be from 0800 to 1800.79
Figure 40 - BBB's time axis has been translated.80
Figure 41 - Schedule plan resulting from previous flight segment manipulations.81
Figure 42 - Daily schedule plan for fleet routing demonstrations. . .	.82
Figure 43 - Answer to the first optimization option.83
Figure 44 - Fleet size input for the fleet routing option.84
Figure 45 - Generic daily ownership cost (doc) input for the fleet routing option.85
Figure 46 - Answer to the second optimization option with fs=1 and doc=-200.86
Figure 47 - Flight of aircraft 001.87
Figure 48 - Answer to the second optimization option with fs=4 and doc=-500.88
Figure 49 - Flights for aircraft in a fleet of four.89
Figure 50 - Flight of one aircraft. AAA-BBB-AAA and CCC-DDD-CCC also qualify.90
Figure 51 - Attempt to delete flight segment 109 leads to an imbalanced flow in the network.91
Figure 52 - After flight segment 109 has been "removed", aircraft 001 is rerouted.92
Figure 53 - The flow bounds on a cycle arc are going to be changed.93
Figure 54 - BBB's cycle arc is selected.94
Figure 55 - BBB's cycle arc lower bound remains 0.95

Figure 56 - BBB's cycle arc upper bound becomes 2.96
Figure 57 - The flow bounds on a flight arc are going to be changed.97
Figure 58 - Flight segment 103 is selected.98
Figure 59 - Flight segment 103's flight arc lower bound remains at 0.99
Figure 60 - Flight segment 103's flight arc upper bound becomes 2.100
Figure 61 - Two aircraft now overnight at BBB.101
Figure 62 - A "shuttle" schedule plan between AAA and BBB.102
Figure 63 - Two aircraft are needed to serve the shuttle.103
Figure 64 - Only one aircraft seems to be required.104
Figure 65 - First part of a SAS schedule plan.105
Figure 66 - Second part of a SAS schedule plan.106
Figure 67 - Six aircraft are needed to serve all flight segments.107
Figure 68 - Answer to the second optimization option with fs=6 and doc=-100.108
Figure 69 - Aircraft flights with fs=6 (first part).109
Figure 70 - Aircraft flights with fs=6 (second part).110
Figure 71 - The aircraft flights have been manually rearranged (first part).111
Figure 72 - The aircraft flights have been manually rearranged (second part)112
Figure 73 - Answer to the second optimization option with fs=5 and doc=-100.113
Figure 74 - Aircraft flights with fs=5 (first part).114

Figure 75 - Aircraft flights with fs=5 (second part).	115
Figure 76 - Aircraft flight with fs=1 (first part).	116
Figure 77 - Aircraft flight with fs=1 (second part).	117
Figure 78 - Change in computation time with number of rotations.	118
Figure 79 - Change in computation time with fleet size.	119
Figure 80 - Input of day of occurrence of a flight segment of a multiple-day schedule plan.	120
Figure 81 - Flight segment 101 has been scheduled for day 14. . . .	121
Figure 82 - Flight segment 101's day of occurrence will be changed.	122
Figure 83 - Flight segment 101 is rescheduled for day 39.	123
Figure 84 - Flight segment 101 has been scheduled for day 39. . . .	124
Figure 85 - Station display for flight segment 101.	125
Figure 86 - One aircraft flies from AAA to FFF in 17 days.	126
Figure 87 - Answer to the first optimization option.	127
Figure 88 - Answer to the second optimization option.	128
Figure 89 - Flight ZZZ-BBB-CCC has been selected, but AAA-BBB-CCC is also possible.	129
Figure 90 - Multiple-day schedule plan between days 1 and 12. . . .	130
Figure 91 - Answer to the second optimization option with fs=3 and doc=-300.	131
Figure 92 - The flights selected for the three aircraft.	132
Figure 93 - The flights selected for two aircraft.	133
Figure 94 - Schedule chart with maintenance arc.	134
Figure 95 - Schedule plan consisting of two flights.	135
Figure 96 - Aircraft are rerouted after a maintenance stopover has been introduced.	136

Figure 97 - Loss of service on a flight segment
when maintenance stopover is too long. 137

Figure 98 - The network data structure consists of a node
linked list and an arc linked list. 138

Contents

1	introduction	11
2	The Out-of-kilter Algorithm	13
2.1	The Minimum Cost Flow Problem	13
2.2	The Out-of-kilter Algorithm	14
2.3	Fleet Routing With The Out-of-kilter Algorithm	18
3	Interactive Dynamic Aircraft Scheduling and Fleet Routing	26
3.1	The IASP for the Daily Schedule Plan	26
3.2	Fleet Routing with the Daily Schedule Plan	30
3.3	Provision for Maintenance Stopovers	34
3.4	The IASP for the Multiple-day Schedule Plan	35
3.5	Fleet Routing with the Multiple-day Schedule Plan	36
4	Conclusions	38
A	Figures	40
B	Data Structure	139

Chapter 1

introduction

The development of a working aircraft schedule plan¹ within an air transportation organization is a formidable task. A schedule plan has to be constructed that not only achieves the goals of geographic, frequency and departure time planning, but also satisfies the constraints of fleet size and maintenance and crew schedules.

The development of a working schedule plan is an iterative process that tries to route aircraft onto flights of a schedule, meet the planning goals and satisfy operational constraints. Presently, the development occurs manually, whereby aircraft are routed on a schedule chart. Since adjustments to the schedule plan or the aircraft routes can require the erasure of existing flights and the addition of new ones onto the chart, the iterative construction of a schedule plan can be time-consuming.

After a working schedule plan has been finalized, many occurrences can require rescheduling of flights and rerouting of aircraft: maintenance and

¹Definitions of schedule-related terms are given on p.18.

crew schedule changes, adverse weather conditions, or new demands for service. This rescheduling is called **dynamic scheduling** and usually occurs under time-pressure.

Because even dynamic scheduling can be time-consuming, the need has been recognized for a computer-based system that would facilitate the development of working schedule plans. However, since the computer is still less versatile than the human mind in making decisions, the development of an efficient man-computer interface has been focused upon [1]. Such an interface has been referred to as a **decision support system (DSS)** [2,3].

In order to gain acceptance, a DSS should, ideally, simulate tried and true schedule construction and modification methods. As part of an evolutionary increase in the effectiveness of a DSS, however, it is desirable to also include operations research models for optimization of fleet routing. In this work, a DSS is presented that combines both features. It employs interactive graphics-based schedule construction and modification tools and optimization routines based on the out-of-kilter algorithm [4,5,6,7] for fleet size determination and fleet routing. The use of the out-of-kilter algorithm for fleet routing was already proposed by Simpson [8] in 1969.

The following chapter covers respectively the theory of the out-of-kilter algorithm, its application to fleet routing, the tools for schedule plan construction and modification, and examples of fleet routing optimization.

Chapter 2

The Out-of-kilter Algorithm

2.1 The Minimum Cost Flow Problem

The minimum cost flow problem is a linear programming problem which has the following formalization:

$$\begin{array}{ll} \text{minimize} & \sum_{ij \in A} c_{ij} x_{ij} \\ \text{subject to} & \sum_{ik \in A} x_{ik} - \sum_{ki \in A} x_{ki} = 0 \quad \forall i \in N \\ & 0 \leq l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall ij \in A \end{array}$$

,where x_{ij} is the amount of flow in arc ij , c_{ij} is the cost a unit of flow incurs in ij , l_{ij} and u_{ij} are respectively the lower and upper bounds on x_{ij} , and A is the set of arcs and N is the set of nodes in the graph of the network.

The optimal solution to the minimum cost flow problem is a set of flows in the arcs of a network which has a minimum total cost. In addition, the flows in this set are conserved at every node of the network and remain between the upper and lower bounds on the flows in the arcs of the network.

The optimal solution to the minimum cost flow problem is obtained when any of the following conditions are satisfied for every arc in the network:

$$\begin{array}{ll} \bar{c}_{ij} = c_{ij} + \Pi_i - \Pi_j < 0 & \text{when } x_{ij} = u_{ij} \\ \bar{c}_{ij} > 0 & \text{when } x_{ij} = \ell_{ij} \\ \bar{c}_{ij} = 0 & \text{when } \ell_{ij} \leq x_{ij} \leq u_{ij} \end{array}$$

, where Π_i is the simplex multiplier for node i . An arc that satisfies one of the above optimality conditions is said to be **in-kilter**, otherwise, it is **out-of-kilter**. Since \bar{c}_{ij} appears as a single variable in the optimality conditions, the **kilter state** of the arc ij can be conveniently represented in a \bar{c}_{ij} vs x_{ij} or **kilter diagram**. The mapping of the kilter states corresponding to the optimality conditions for arc ij then results in a **kilterline** of the form shown in Figure 1 (see Appendix A).

2.2 The Out-of-kilter Algorithm

An algorithm has been developed that modifies the arc flows and node multipliers in order to bring every arc of the network **in-kilter**; it is called the **out-of-kilter algorithm**.

The out-of-kilter algorithm requires an initial set of values for the arc flows and the node multipliers; usually, all of these values are equal to zero, but this need not be the case. Indeed, the values of the arc flows and node multipliers of a problem that differs only slightly from the one studied are also acceptable. After the arc flows and node multipliers have been

initialized, the algorithm searches for all out-of-kilter arcs in the network and attempts to bring them in-kilter, one by one. There are two algorithms in the out-of-kilter algorithm which can accomplish this task: one that adjusts the arc flows and one that modifies the node multipliers.

The former algorithm is a labelling algorithm that seeks to change the flow in an arc ij that is out-of-kilter. Depending on the value of \bar{c}_{ij} , it either increases or decreases the flow from i to j . In both instances, a flow-augmenting path from respectively j to i or i to j has to exist in the network in order for the flow to be conserved at i and j . The node labels - i and j - are by convention reversed in the first case, so that the flow augmenting path always goes from i to j .

The arcs that are included in the flow-augmenting path can be forward or reverse. A forward arc is included in the path if either

$$\begin{aligned} &\bar{c}_{ij} > 0 \quad \text{and} \quad x_{ij} > \ell_{ij} \\ \text{or} \quad &\bar{c}_{ij} \leq 0 \quad \text{and} \quad x_{ij} > u_{ij} \end{aligned}$$

and a reverse arc is added if either

$$\begin{aligned} &\bar{c}_{ij} < 0 \quad \text{and} \quad x_{ij} > u_{ij} \\ \text{or} \quad &\bar{c}_{ij} \geq 0 \quad \text{and} \quad x_{ij} > \ell_{ij}. \end{aligned}$$

These acceptable kilter states are shown in Figure 2.

Since no arc that is already in-kilter should be brought out-of-kilter and no distance between an out-of-kilter arc's position in the kilter diagram and its kilter line should be increased, flow changes of too great a magnitude in the arcs of the flow-augmenting path are prohibited. To this end, the labelling algorithm assigns two labels to every new node of the path. One label is the label of the node that precedes the new node in the path from i (with, in addition, a $+$ or $-$ that indicates whether the new arc is forward or reverse). The other label is an integer whose value is the maximum additional amount of flow that can be sent along the path without moving any arc in the path beyond its kilter line. This flow increment is the smaller of the previous flow increment and the maximum flow increment possible in the new arc of the path. The new arc's maximum flow increment equals:

$$\begin{aligned}
 \text{for a forward arc } ij : \quad & l_{ij} - x_{ij} \quad \text{when } \bar{c}_{ij} > 0 \\
 & u_{ij} - x_{ij} \quad \text{when } \bar{c}_{ij} \leq 0 \\
 \text{for a reverse arc } ij : \quad & x_{ij} - l_{ij} \quad \text{when } \bar{c}_{ij} \geq 0 \\
 & x_{ij} - u_{ij} \quad \text{when } \bar{c}_{ij} < 0.
 \end{aligned}$$

When the endnode j of the out-of-kilter arc ij has been labelled, a flow-augmenting path has been found (a breakthrough has occurred) and the value of j 's flow increment label is added to the flow in every arc of the flow augmenting path. Hereafter, all label pairs of the nodes of the path are discarded and the kilter state of ij is determined once more. The labelling algorithm is applied to ij until ij is either in-kilter or no flow-augmenting path is found (a non-breakthrough has occurred).

In the latter case, the second algorithm is employed to aid in bringing the out-of-kilter arc in-kilter. This algorithm changes the value of the node multipliers of the labelled nodes whose incident arcs link labelled and unlabelled nodes by an amount M which is the minimum of:

- The minimum of \bar{c}_{ij} for all forward arcs ij linking labelled and unlabelled nodes for which $\bar{c}_{ij} > 0$ and $l_{ij} \leq x_{ij} < u_{ij}$.
- The minimum of $|\bar{c}_{ij}|$ for all reverse arcs ij linking labelled and unlabelled nodes for which $\bar{c}_{ij} < 0$ and $l_{ij} < x_{ij} \leq u_{ij}$.
- Infinity.

The first two cases are shown in Figure 3. The new values of the reduced costs of the forward and reverse arcs will therefore be, respectively, $\bar{c}_{ij} - M$ and $\bar{c}_{ij} + M$. As a result, at least one arc's new position in its kilter diagram will be on the horizontal segment of the kilter line and no arc's position will be beyond this segment. If M has an infinite value, however, no feasible flow exists and the out-of-kilter algorithm then terminates. If infeasibility has not occurred, the second algorithm will have prepared at least one more node for labelling by the first algorithm - thereby extending the flow-augmenting path - or the out-of-kilter arc will have been brought in-kilter. In the latter case, the out-of-kilter algorithm searches for a new out-of-kilter arc; otherwise, it restarts the labelling process from node i , since changes in the node multipliers may have affected the effective capacity of the flow-augmenting path.

Eventually, the out-of-kilter algorithm either has brought all out-of-kilter arcs in-kilter, in which case a minimum cost flow has been found, or has not been able to find a flow augmenting path for an out-of-kilter arc, in which case no feasible flow can exist in the network, due to a conflict between flow constraints.

2.3 Fleet Routing With The Out-of-kilter Algorithm

The out-of-kilter algorithm can be applied to various network-related mathematical models of realistic problems when the cost parameters of the minimum cost flow objective function are changed, or arcs or arc parameters are added to the original network. Consequently, the out-of-kilter algorithm can solve some of the most common network flow problems: maximum flow, shortest path, assignment, transportation and transshipment problems. This fact clearly demonstrates its power and usefulness.

On the following pages, the application of the out-of-kilter algorithm to mathematical models of two common and closely related air transportation problems is discussed:

- Find the minimum size of an aircraft fleet, so that all flight segments of the schedule plan are served.
- Find the optimal flights (which accommodate the flight segments of highest "value") over a schedule plan for an aircraft fleet of fixed size.

A flight segment of a schedule plan is a non-stop connection between two airfields at a particular date and time and a flight represents the sequence of flight segments assigned to a single aircraft for service; the geographic trace of a flight is called a route. The schedule plan is then a collection of flight segments that are to be served, or can be served, within a planned time horizon.

Two types of schedule plans will be converted into mathematical models that are suitable for solution with the out-of-kilter algorithm. The first type of schedule plan, the daily schedule plan, consists of flight segments that are to be served on a daily basis and whose flights (rotations) usually¹ originate and terminate at the same airfield, where the aircraft will overnight. The second type of schedule plan, the multiple-day schedule plan, contains flight segments that have been planned for service during a period of at least one day. The aircraft of a fleet assigned to this schedule plan can therefore overnight at stations along their route and do not have to return to the station at which their flight originated.

The value of a flight segment can have several meanings, depending on the nature and circumstances of the flight segment or the objectives of the people who have an interest in it. The value of a commercial flight segment, for example, might be its revenue, whereas the value of a military flight segment might be directly related to the priority of its cargo.

¹An exception is presented on p.31.

One important restriction to the application of the models is that neither one can satisfy the optimization criteria for a fleet of aircraft of mixed type. A fleet routing problem for a mixed-type aircraft fleet will therefore have to be decomposed into subproblems according to each aircraft type.

The type of schedule plan determines the form of its equivalent network, which in turn influences the identity of the mathematical model. Arcs, nodes, lower and upper bounds and flow costs on the arcs are the building blocks of a network that corresponds to a schedule plan. Flight, ground and cycle arcs and station event nodes provide the framework of the network. Here, a station event corresponds to an arrival² or departure of a flight segment and a ground arc connects two successive station events. In the case of a daily schedule plan, the cycle (overnight) arc links the last station event to the first at each station in the schedule plan. The arcs and nodes of this schedule plan can be graphically represented as a schedule chart, as shown in Figure 4. In the case of the multiple-day schedule plan, the network also contains arcs that connect the "first" station events with a source node and arcs that connect the "last" station events with a sink node. One cycle arc connects the sink node with the source node and there are no overnight arcs in this network, which is shown in Figure 5.

The optimization problem and the type of schedule plan determine the mathematical model and the values of the parameters of the arcs in the network. The arc parameters are presented in Figure 6 and the mathematical model formulations in Figure 7.

²In practice, the arrival time is converted into an earliest time of departure, or "ready" time.

The values of the arc parameters for the first optimization problem - determining the minimum fleet size - are identical for the daily and multiple-day schedule plans. The "value" of a flight arc is negative, reflecting its revenue or negative cost. Since the first optimization problem is actually reduced to the counting of the minimum number of aircraft needed to serve all flight segments of the schedule plan, both the lower and upper bound on the flow in the flight arcs are equal to one, so that exactly one aircraft will serve each flight segment. As is shown in Figure 7, for the daily schedule plan network the sum of all cycle arc flows equals the fleet size: $\sum_{ij \in A} c_{ij} x_{ij} = \sum_{ij \in \text{cycle arcs}} x_{ij}$. For the multiple-day schedule plan network, no overnight arcs exist and the fleet size is consequently equal to the flow in the cycle arc ts , x_{ts} .

The mathematical models for the second optimization problem - finding the optimal flights in a schedule plan for an aircraft fleet of fixed size - are not as straightforward as the previous mathematical models; a few concepts need to be introduced before they can be explained:

-The daily ownership cost of an aircraft, doc , is a measure of the daily expense associated with the ownership and operation of an aircraft; it has a positive value.

-The take of an aircraft: $\sum_{ij \in \text{flight arcs}} c_{ij} x_{ij}$, where ij is an arc in the network representing a flight segment of the aircraft's flight.

-The net take of an aircraft: $\sum_{ij \in \text{flight arcs}} c_{ij} x_{ij} + doc$

-The net take of an aircraft fleet is the sum of the takes of all aircraft in the fleet:

-for the daily schedule plan: $\sum_{ij \in \text{flight arcs}} c_{ij} x_{ij} + doc \left(\sum_{ij \in \text{cycle arcs}} x_{ij} \right)$

-for the multiple-day schedule plan: $\sum_{ij \in \text{flight arcs}} c_{ij} x_{ij} + doc x_{ts}$

-The maximum doc is that doc for which the net take of the aircraft fleet becomes zero.

For the second optimization problem, the objective is to maximize the net take of the aircraft fleet. In order to describe the mathematical models in minimum cost form, the objective functions in Figure 7 are the negative of the fleet net takes given above. Both mathematical models now contain the additional constraint that the fleet size cannot exceed a given number, FS . As is shown in Figure 6, the lower and upper bounds on the flow in the cycle arcs of the daily schedule plan network, l_c and u_c , can be set to values that reflect restrictions imposed on the number of aircraft that can overnight at the airfields. The default values are respectively 0 and ∞ , so that an indefinite number of aircraft can overnight at each airfield. The lower and upper bounds on the flight arcs, l_f and u_f , can also be changed, so that several aircraft can simultaneously serve a given flight segment. For the multiple-day schedule network, u_c determines the maximum fleet size, while the flight arc flow bounds allow for not more than one aircraft to serve each flight segment. For both networks, the daily ownership cost is assigned to the cycle arc(s).

As was described above, the solution to the first optimization problem consists of the minimum fleet size necessary to serve all flight segments of the schedule plan. It is obtained by employing the out-of-kilter algorithm once and summing the flows in all the cycle arcs.

The solution to the second optimization problem consists of the fleet size and the flights assigned to each aircraft in the fleet. The solution method for the fleet size determination is based upon the fact that the net take of each aircraft in the fleet has to be positive; if there are aircraft whose take is less than the daily ownership cost, they are to be removed from the aircraft fleet and the aircraft in the remainder of the fleet can then be assigned new flights, which may include individual flight segments of the discarded flights if these increase their take. Since the daily ownership cost thus indirectly sizes the aircraft fleet, it assumes the role of control variable in the iteration procedure that maximizes the fleet's net take, while satisfying the fleet size constraint.

The iteration procedure arises from the use of a Lagrange multiplier technique and is explained with the aid of the following scenario. Suppose that an initial doc , doc_1 , results in a fleet size fs , larger than FS , as shown in Figure 8. This situation can occur when FS is less than the fleet size necessary to serve all flights in the schedule plan. Since fs decreases with increasing doc , doc_1 is added to doc until fs is less than or equal to FS . At doc_3 , fs is less than FS and doc_3 is therefore decreased by an amount equal to one-half of the original increase in an attempt to reach the FS mark.

Since fs is still less than FS at $doc4$, $doc4$ is again decreased, so that $doc2$ is reached anew. Now, however, only one-quarter of the original increase (one-half of the previous decrease) is added to $doc2$. In this case, the iteration procedure terminates at $doc5$ because the termination criterion is now satisfied: fs is equal to FS .

If the aircraft fleet is larger than is necessary to serve all flight segments in the schedule plan, it is decreased to the maximum fleet size for the given schedule plan before the iteration procedure is started. In section 3.2, it is shown that this procedure results in a significant reduction in computation time, because the solution is obtained after fewer iterations.

Upon termination of the iteration procedure described above, the out-of-kilter algorithm has determined the fleet size and the set of flight segments of the schedule plan that are served, but not the aircraft flights. The latter are determined in a final employment of the out-of-kilter algorithm. First, the arc flows and node multipliers are reinitialized and the cycle arc costs are set to the final value of doc . Then, as each arc of the network is brought in-kilter for the first time, the set of flight segments that are brought in-kilter along with it are assigned a unique label. The time-ordered sequence of those flight segments then forms the flight of a single aircraft in the fleet.

In this section, the determination of the minimum fleet size and the set of flights corresponding to a given fleet size was presented as a pair of optimization problems for two schedule plans, a daily and a multiple-day schedule plan. A solution to these problems with the out-of-kilter algorithm

led to the conversion of the optimization problems and schedule plans into four distinct mathematical models. Whereas the solution to the former optimization problem could be determined after a single employment of the out-of-kilter algorithm, the latter optimization problem dictated an iterative solution to the fleet size constraint, based on the variation of the generic daily ownership cost of the aircraft in the fleet. Aircraft flights came into existence as batches of flight segments simultaneously brought in-kilter.

In the following chapter, the working of the schedule planning/fleet routing DSS will be discussed.

Chapter 3

Interactive Dynamic Aircraft Scheduling and Fleet Routing

The PASCAL procedures that collectively solve both optimization problems exist as two sets, one for the daily schedule plan and one for the multiple-day schedule plan, and each set is imbedded within a copy of an interactive aircraft scheduling program (IASP) that has been developed under Version 3 of the Apple Macxl PASCAL Workshop. The features of the IASP and the optimization problem solvers will be described hereafter. Appendix B explains the data structure behind the networks that represent the schedule plans.

3.1 The IASP for the Daily Schedule Plan

The IASP allows for the dynamic construction and modification of a schedule plan because menu options can be moused to activate PASCAL procedures that add, delete, move, or modify flight segments displayed on a schedule chart; presently, a maximum of 104 flight segments can be created.

Figures 9 through 22 animate the construction of a simple daily schedule plan and the alteration of flight segment characteristics.

Upon selection of the add option, the IASP prompts for the flight segment's number, origin code name, destination code name, value and duration, as may be seen in Figures 9 through 13¹. After this information has been input, a rectangle that depicts the flight segment appears and can be moved with the mouse to the desired position on the schedule chart. For any particular departure time, the vertical position on the schedule chart is unimportant. The minutes past the hour of the departure time are displayed in the upper-righthand corner, as shown in Figure 14. Figure 15 presents a flight segment that departs from AAA at 2 pm and arrives at BBB at 5 pm and has a value of 100.

In order to alter any flight segment characteristic, both the modify option and the flight rectangle must be moused; a new display then appears as presented in Figure 16. If, for example, the flight segment's value is to be increased to 200, the value bar is moused and the new value is input, as is shown in Figure 17.

Figure 18 presents a schedule plan consisting of four flight segments between stations AAA, BBB and CCC. Any flight segment can be moved or deleted by mousing both the corresponding option and the flight segment rectangle itself. The result of selecting each option is shown in Figures 19 through 22; flight segment 104 is first moved and then deleted.

¹ Although the cost parameter of a flight segment is negative, the value of the flight segment is input as a positive number, corresponding to the inflow of "value".

The display can also be translated along the time axis of the schedule chart. To this end, the time axis option is moused and the "time rectangle" is moved; only hourly translations can be made. The time axis option display is shown in Figure 23 (the time interval of the portion of the schedule chart currently displayed is painted black) and the display resulting from the translation of the time rectangle is shown in Figure 24.

Mousing the schedule option results in a tabular display of the schedule plan, which may be seen in Figure 25.

With the station option, all arrivals and departures at two airfields of the schedule plan can be seen. In addition, new flight segments between those airfields can be added and existing flight segments can be deleted and moved. The modify, schedule and end options all return the display to the original schedule chart.

Figure 26 presents the station display for stations AAA and BBB. In order to delete a flight segment, the delete option is selected and the mousebutton is pressed when the mouse arrow is positioned over the flight segment number of either the arrival or departure arrow, as shown in Figures 27 and 28.

In order to add a flight segment, its number, duration and value are input and its arrival or departure time is selected by positioning the mouse arrow in the desired "time field" and mousing at the desired time. This

sequence of events is shown in Figures 29 through 33. Flight segment 107 departs from AAA at 1530 and arrives at BBB at 1730. In Figure 34, the four time fields are indicated; mousing in any one is sufficient to schedule a flight segment between AAA and BBB. If the departure time is DT and the arrival time is AT, then mousing in time field

- 1 fixes the AT of BBB to AAA,
- 2 fixes the DT of AAA to BBB,
- 3 fixes the AT of AAA to BBB,
- 4 fixes the DT of BBB to AAA.

A flight segment's arrival or departure time can be moved by selecting the slide option, marking the arrival or departure arrow, as before, and mousing at the new arrival or departure time, as shown in Figures 35, 36 and 37, where the departure time of flight segment 101 has been moved to 1410.

With the time axis option, the visible portion of each station's time window can be changed independently. It suffices to select the desired station option and move the time rectangle to its new position, as was explained before. In Figures 38, 39 and 40, station BBB is selected and the visible portion of its time axis is translated.

As a result of the flight segment manipulations described above, the schedule chart display now appears as shown in Figure 41.

The IASP offers an array of tools with which a schedule plan can be constructed and modified. It also employs the schedule plan as an input to both optimization options. Hereafter, the input to and output from both optimization options will be described and evaluated for the daily schedule plan.

3.2 Fleet Routing with the Daily Schedule Plan

The input-output features of the optimization options will be identified with the aid of the schedule plan presented in the form of the schedule chart of Figure 42.

Upon selecting the first optimization option (1), the display becomes as given in Figure 43. Four aircraft are needed to serve all flight segments and they overnight at stations AAA, BBB, and DDD. The response time of .2 sec demonstrates the high speed with which the network is constructed and the minimum fleet size problem is solved.

The second optimization option (2) requires as input the maximum number of aircraft in the fleet and the daily ownership cost of the aircraft, in addition to the schedule plan. As is shown in Figures 44 and 45, the flight of one aircraft with a doc of -200 is to be found ². The first part of the solution is presented in Figure 46. The aircraft's net take is $360 - 200 = 160$,

²Again, the cost parameter of a cycle arc is positive, but the doc is input as a negative number to account for the outflow of "value".

its maximum doc is 360 and it overnights at station AAA. The message "grey flights are flown" points to the second part of the solution, which can be seen after mousing. The grey (black here) flight segment rectangles that are tagged with the aircraft number belong indeed to the flight aircraft 001 can serve, as may be seen in Figure 47. Aircraft 001's take is the highest take a single aircraft can achieve for this schedule plan. Figures 48 and 49 confirm the answer of the first optimization option: four aircraft are needed to serve all flight segments in the schedule plan. Since the doc was set at -500, which is higher than the maximum doc of -265, the fleet net take is negative. Although flight DDD-CCC-DDD has the same total value as flight BBB-CCC-BBB, the program assigns one aircraft to each flight. As a result, flights that are not included in the final schedule plan, but could be preferable to flights of equal total value that are included, should be checked for. An example is given in Figure 50. If only one aircraft is available, the program selects flight EEE-FFF-EEE, but either one of the other two flights may be preferable.

Under realistic conditions, some flight segments may become unservicable at certain moments. Rather than deleting from the schedule plan the flights whose flight segments cannot all be served, the IASP offers the opportunity to "remove" those flight segments and reoptimize the fleet routing for the modified schedule plan. An obvious step would be to employ the delete option to this end; however, if, for example, flight segment 109 of Figure 47 is thus deleted, the display becomes as shown in Figure 51, because of a violation of the flow conservation constraint at stations CCC and EEE: their number of arrivals does not equal their number of departures.

Instead of deleting the non-servicable flight segments, the modify option should be employed; it suffices to select the value bar and input "out" to "remove" the flight segment from the schedule plan³. The flight segment will then be displayed in light gray on the schedule chart. If flight segment 109 has become unservicable, a single aircraft will be rerouted from the optimal flight shown in Figure 47 to the one shown in Figure 52.

With the u,l option, the flow bounds on any flight arc or cycle arc can be adjusted. If the number of overnight aircraft at an airfield has to be restricted, the word "stat" (short for station) and the station code name and the upper and lower bounds have to be input, as shown in Figures 53 through 56 for station BBB, where the upper bound is set to 2. The number of aircraft that can simultaneously serve a flight segment can be changed by respectively inputting "ft", selecting the flight segment rectangle and inputting the upper and lower bounds on the flow in its corresponding flight arc, as shown in Figures 57 through 60 for flight segment 103. If both flight segment 103 and 104 have an upper bound equal to 2 and the fleet size is set to 5, the solution to option 2 is as shown in Figure 61. Two aircraft now overnight at station BBB and serve flight segments 103 and 104.

Figure 62 presents a "shuttle" schedule plan between stations AAA and BBB. The solution to option 2 reveals that two aircraft are needed to serve both flight segments, as shown in Figure 63 (the doc was -100). In Figure 64, however, all flight segments have been labelled with '001'. The reason is that they were all brought in-kilter simultaneously and therefore labelled

³"in" reinserts the flight segment into the schedule plan.

accordingly. For any schedule plan, the presence of "shuttles", if not immediately visible on the schedule chart, will always be noticeable from the discrepancy between the actual fleet size and the maximum label number of the flight segments in the schedule plan.

The schedule plan discussed above has served to explain the characteristics of the IASP and the optimization options; a final, somewhat more realistic schedule plan, will now be presented. This schedule plan will also serve as a test case for the performance measurement of the optimization algorithms for the daily schedule plan. The schedule plan is shown in Figures 65 and 66 and has been derived from a portion of an actual Scandinavian Airline System flight schedule. Each row of flight segments represents a complete rotation. Since there are six rotations, six aircraft should be required in order to serve all flight segments of the schedule plan. The solution to option 1 confirms the fleet size and the overnight stations and is shown in Figure 67; it was found after .8 sec. The solution to option 2 - with an input of six aircraft for the fleet size and -100 for the doc - also confirms the above result, but includes rotations different from the ones that were apparent. This solution is presented in Figures 68, 69 and 70. The flight segments belonging to the same aircraft can now be manually rearranged so that they appear on the same horizontal line, as shown in Figures 71 and 72. The fleet net take for an aircraft fleet that flies the original rotations or the ones found by the optimization procedures will naturally be the same; if, however, one aircraft breaks down, it may not be obvious which rotations should be served so as to minimize the loss in take; the optimization algorithms will provide that answer. For the solution shown in Figures 73,

74 and 75, the fleet size was set to five, while the doc was kept at -100. The fleet net take has decreased to 3815, which is attributable to the unservicable flight. It is worth noting that the flights shown in Figures 74 and 75 are different from those in Figures 69 and 70 because the reoptimization has reallocated the flight segments to the aircraft of the fleet. If only one aircraft was available, it would serve the flight shown in Figures 76 and 77.

The SAS schedule plan example has been used to determine the performance of the software. Figure 78 shows that the computation time grows linearly with the schedule plan size. Figure 79 shows the variation of computation time with fleet size for a fixed schedule plan. The computation time for fleet sizes smaller than the maximum fleet size (6) varies from two to three times the computation time for the maximum fleet size case, because the number of iterations required to obtain a solution varies with the fleet size. The computation time for fleet sizes larger than the maximum fleet size is, however, constant and approximately equal to the minimum computation time, because the fleet size is automatically decreased to the maximum fleet size before the iteration is started.

3.3 Provision for Maintenance Stopovers

Maintenance checks, regular or unexpected, are performed on all operating aircraft. They are represented as maintenance arcs in the daily schedule plan network, which force aircraft to undergo maintenance at a particular station and time. A maintenance arc is depicted in Figure 94; the cost of the arc is set to zero because maintenance accounting is normally

distinguished from flight service accounting. The addition of a maintenance arc should not change the flow in the arcs whose finish time is less than the start time of the maintenance arc if an out-of-kilter flow solution already existed for the original network, this to avoid rerouting of the aircraft prior to the maintenance stopover.

Figures 95 and 96 demonstrate the rerouting that can occur after a maintenance stopover that has been introduced into a schedule plan. In Figure 95, the flights of two aircraft have been determined with the second optimization option. One flight includes flight segments 101 and 102, the other includes flight segments 103 and 104. In Figure 96, a maintenance stopover of 3 hours at station BBB - depicted by segment M01 - was added and the flow bounds of segments 101 and M01 were set to 1 with the u,l option. All flight segments are still served, but the flights have changed. Whereas one flight now consists of flight segments 102 and 103, the other consists of flight segments 101 and 104 and the maintenance stopover at station BBB.

Figure 97 shows that aircraft 001 is not able to serve flight segment 104 if the maintenance stopover lasts until a time later than the departure time of the flight segment.

3.4 The IASP for the Multiple-day Schedule Plan

The flight segments of a multiple-day schedule plan are distinguished,

not only by the characteristics they have in common with the daily schedule plan flight segments, but also by their day of occurrence. With this program, flight segments can be scheduled within a time horizon of 999 days. The day for which the flight segment is scheduled simply has to be input as part of the input of the flight segment characteristics, as shown in Figure 80. In Figure 81, flight segment 101 has been scheduled for day 14. If it has to be rescheduled, the day option should be selected and a new day input, as demonstrated in Figures 82, 83 and 84, where flight segment 101 has been rescheduled for day 39.

Figure 85 presents the station display for flight segment 101. The day for which the flight segment is scheduled now marks both the arrival and departure arrows.

3.5 Fleet Routing with the Multiple-day Schedule Plan

Figures 86, 87 and 88 show the solution to respectively option 1 and option 2 for a flight scheduled for one aircraft that has a doc of -100. The flight starts at station AAA at day 12 and ends 17 days later at station FFF. The overnight stations of the flight are not explicitly shown, but can be inferred from the days of occurrence of successive flight segments.

When two aircraft compete for the same flight segment and the take of either aircraft would be the same, the program assigns the flight segment to one aircraft, as shown in Figure 89, where the fleet size was set to one.

Flight segments that are not included in the final solution, but may be preferable, should therefore be inspected.

Figure 90 presents a schedule plan set to occur between days 1 and 12. The schedule plan consists of three flights, as shown in Figures 91 and 92 (the doc was -100) . The solution was obtained after barely .3 sec. If only two aircraft were available, their flights would become as indicated in Figure 93. The flight of aircraft 001 is now different from the one of Figure 92 because the aircraft had access to the flight segments of the "third" aircraft.

In this chapter, schedule construction and modification with the IASP and fleet routing with imbedded optimization procedures based upon the out-of-kilter algorithm have been described. It has been shown that a schedule plan is easily built and modified and that optimal flights and minimal fleet sizes are, in most cases, quickly obtained. Maintenance stopovers, if carefully scheduled, can be integrated into a schedule plan with a minimal loss of regular service. The usefulness of the solutions to the optimization problems, however, ultimately have to be considered relative to constraints and demands other than those related to the schedule plan itself. This fact will be expanded upon in the conclusion presented hereafter.

Chapter 4

Conclusions

Computer-based schedule plan construction and modification tools, in combination with out-of-kilter-based fleet routing optimization functions create an effective decision support system for dynamic scheduling. Efficient dynamic scheduling is achieved because of the straightforward interaction between keyboard and mouse inputs and graphically displayed information.

The fleet routing optimization procedures provide answers quickly, but their quality and applicability has to be assessed:

- Flight segments not included in the solution may be preferable to flight segments of equal value that are included.
- Schedule plan or fleet size changes may lead to a complete reconfiguration of the aircraft flights. It would be necessary to reconfigure maintenance and crew schedules accordingly.
- The proposed overnight stations in a daily schedule plan may not be desirable, but constraints can be imposed to produce the desired overnighing.

The need for assessing the quality of fleet routing optimization solutions is unavoidable and defines and restricts the role of a decision support system. This does not represent a real drawback, however, for many scheduling scenarios can be quickly staged and interpreted in succession, so that a realistic solution can be found which is fairly close to the "blind" optimum. Since many alternatives can be evaluated, such a solution will certainly be better than one that is obtained manually within the same time-frame.

The functions of the DSS presented in this work can be expanded and refined. For example:

- The manual rearrangement of flight segments on the schedule chart according to aircraft number can be automated.
- The input of maintenance segment characteristics can be separated from the input of the flight segment characteristics; flow bounds on the maintenance arcs can then be set automatically. The fixation of flows in segments that finish at times before a given time can also be automated.
- Software that correctly labels the flights of aircraft for shuttle plans can be developed.

The software of this DSS, written in PASCAL, should be easily transferrable to computer systems with more advanced graphics features and larger database capacities. In addition, this DSS should also be useful to organizations other than those in the air transportation field, because the segment rectangles and their characteristics can be easily redefined.

Appendix A

Figures

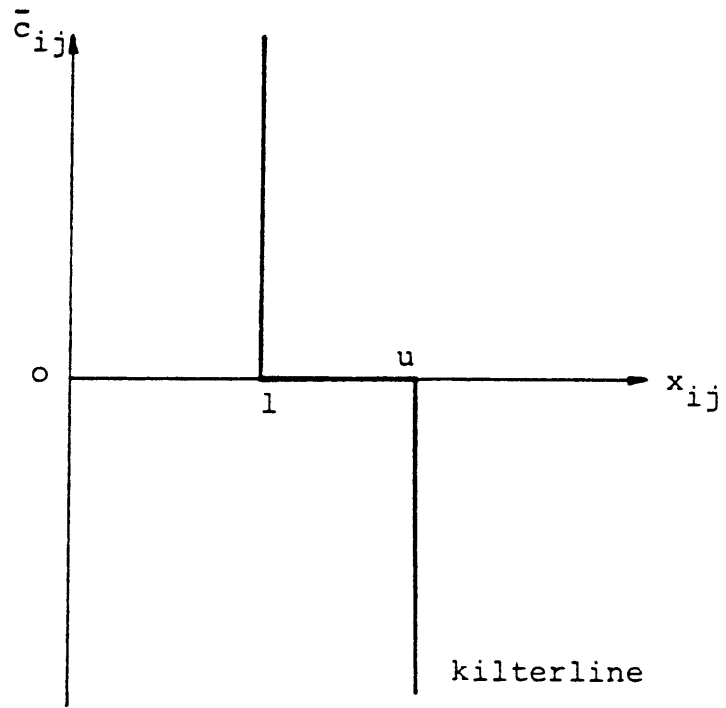


Figure 1 - The kilter diagram of arc ij

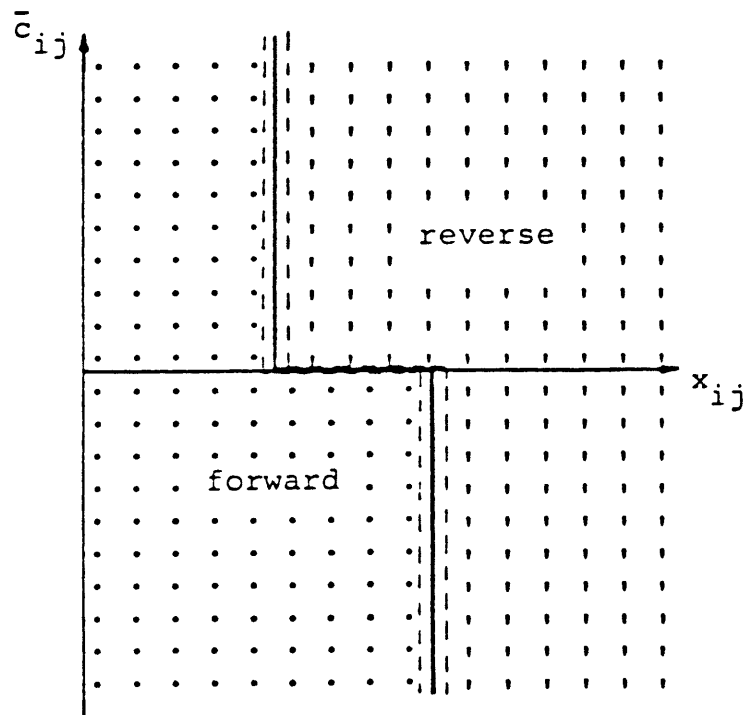


Figure 2 - Acceptable kilter states for forward and reverse arcs to be included in the flow-augmenting path

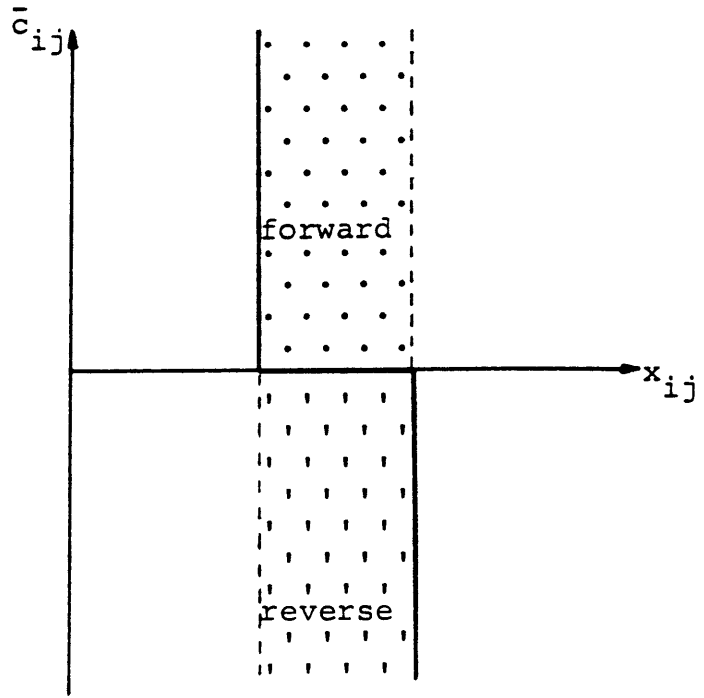


Figure 3 - Kilter states of arcs linking labelled and unlabelled nodes of which the node multipliers will be changed

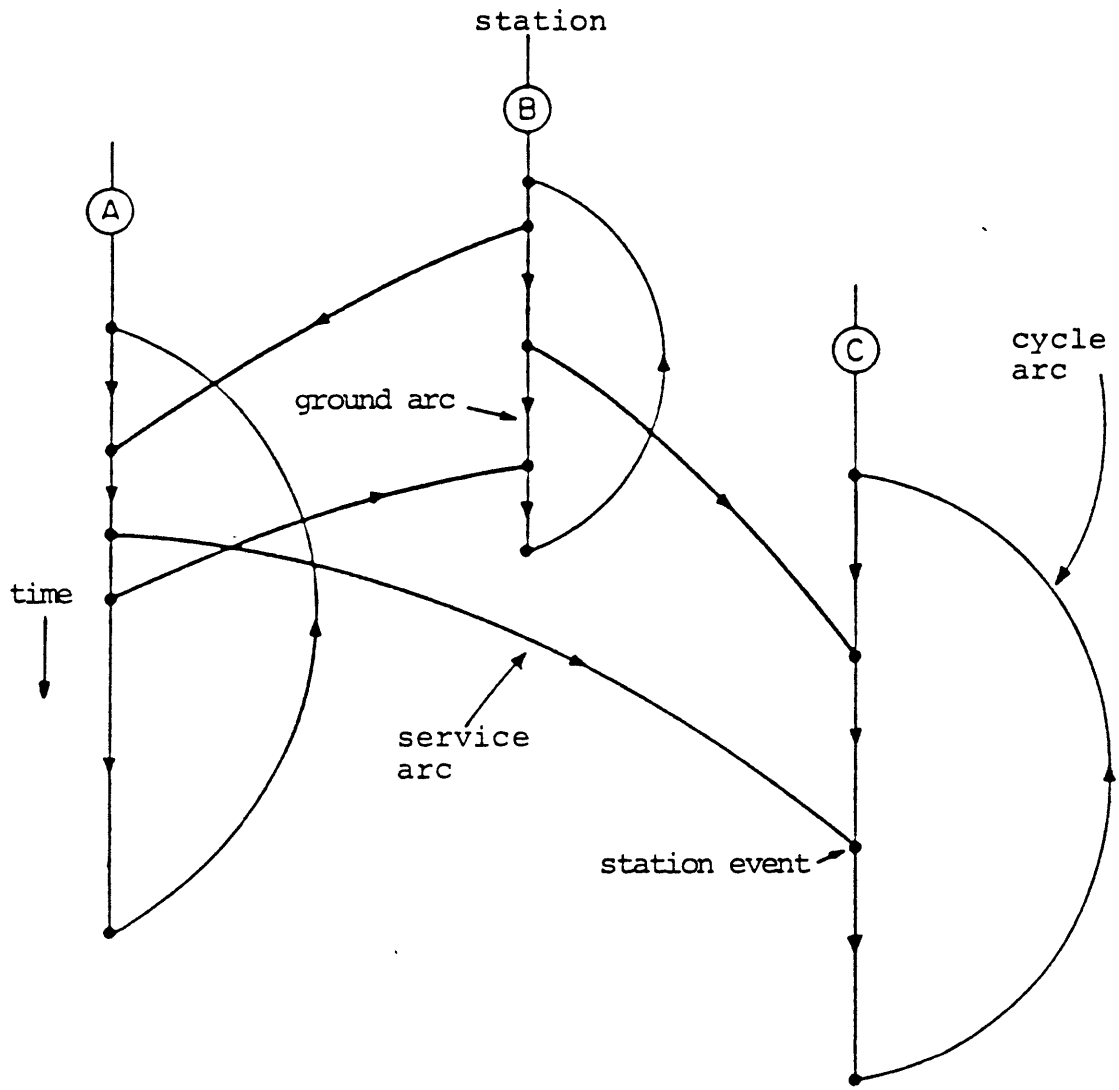


Figure 4 - Daily schedule plan schedule chart

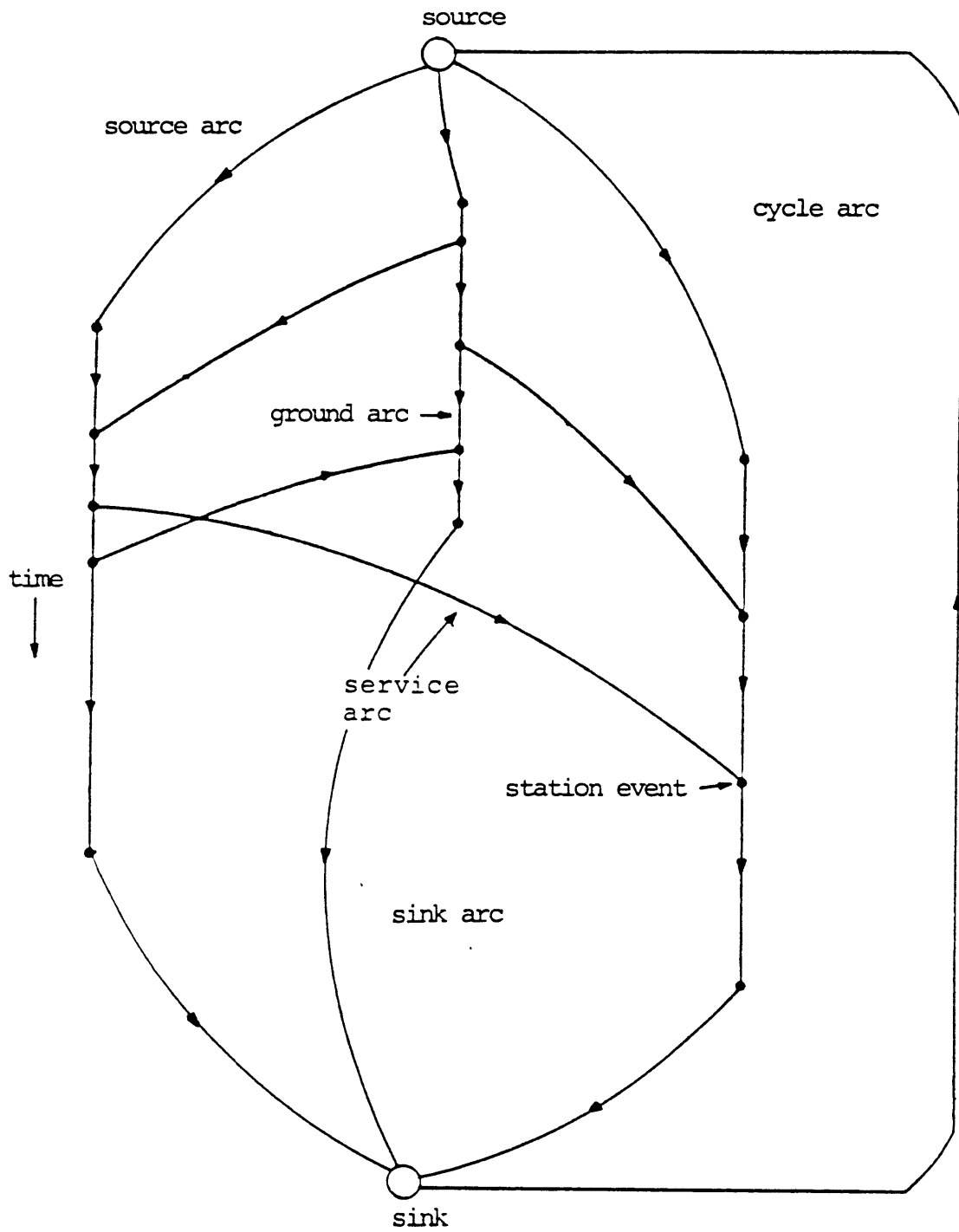


Figure 5 - Multiple-day schedule plan schedule chart

Optimization Problem		Minimum Fleet Size	Fleet Routing
Schedule Plan			
Daily	ground arcs:	$l = 0, u = \infty, c = 0$	$l = 0, u = \infty, c = 0$
	cycle arcs:	$l = 0, u = \infty, c = 1$	$l = l_c(0), u = u_c(\infty), c = doc$
	flight arcs:	$l = 1, u = 1, c = value$	$l = l_f(0), u = u_f(1), c = value$
Multiple-day	ground arcs:	$l = 0, u = \infty, c = 0$	$l = 0, u = \infty, c = 0$
	cycle arcs:	$l = 0, u = \infty, c = 1$	$l = 0, u = u_c, c = doc$
	flight arcs:	$l = 1, u = 1, c = value$	$l = 0, u = 1, c = value$
	source,sink arcs:	$l = 0, u = \infty, c = 0$	$l = 0, u = \infty, c = 0$
			l =lower bound u =upper bound c =cost doc =daily ownership cost

Figure 6 - Network arc parameters by schedule plan and optimization option

Optimization Problem

Minimum Fleet Size

Fleet Routing

Schedule Plan

Daily

$$\begin{aligned} \min \quad & \sum_{ij \in \text{cycle arcs}} x_{ij} \\ & \sum_{ik} x_{ik} - \sum_{ki} x_{ki} = 0 \quad \forall i \in N \\ & 0 \leq l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall ij \in A \end{aligned}$$

$$\begin{aligned} \min \quad & -[\sum_{ij \in \text{flight arcs}} c_{ij} x_{ij} + \text{doc} (\sum_{ij \in \text{cycle arcs}} x_{ij})] \\ & \sum_{ij \in \text{cycle arcs}} x_{ij} \leq FS \\ & \sum_{ik} x_{ik} - \sum_{ki} x_{ki} = 0 \quad \forall i \in N \\ & 0 \leq l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall ij \in A \end{aligned}$$

Multiple-day

$$\begin{aligned} \min \quad & x_{ts} \\ & \sum_{ik} x_{ik} - \sum_{ki} x_{ki} = 0 \quad \forall i \in N \\ & \sum_{si} x_{si} - x_{ts} = 0 \\ & \sum_{it} x_{it} - x_{ts} = 0 \\ & 0 \leq l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall ij \in A \end{aligned}$$

$$\begin{aligned} \min \quad & -[\sum_{ij \in \text{flight arcs}} c_{ij} x_{ij} + \text{doc } x_{ts}] \\ & x_{ts} \leq FS \\ & \sum_{ik} x_{ik} - \sum_{ki} x_{ki} = 0 \quad \forall i \in N \\ & \sum_{si} x_{si} - x_{ts} = 0 \\ & \sum_{it} x_{it} - x_{ts} = 0 \\ & 0 \leq l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall ij \in A \end{aligned}$$

s = source
t = sink

Figure 7 - Mathematical formulations by schedule plan and optimization option

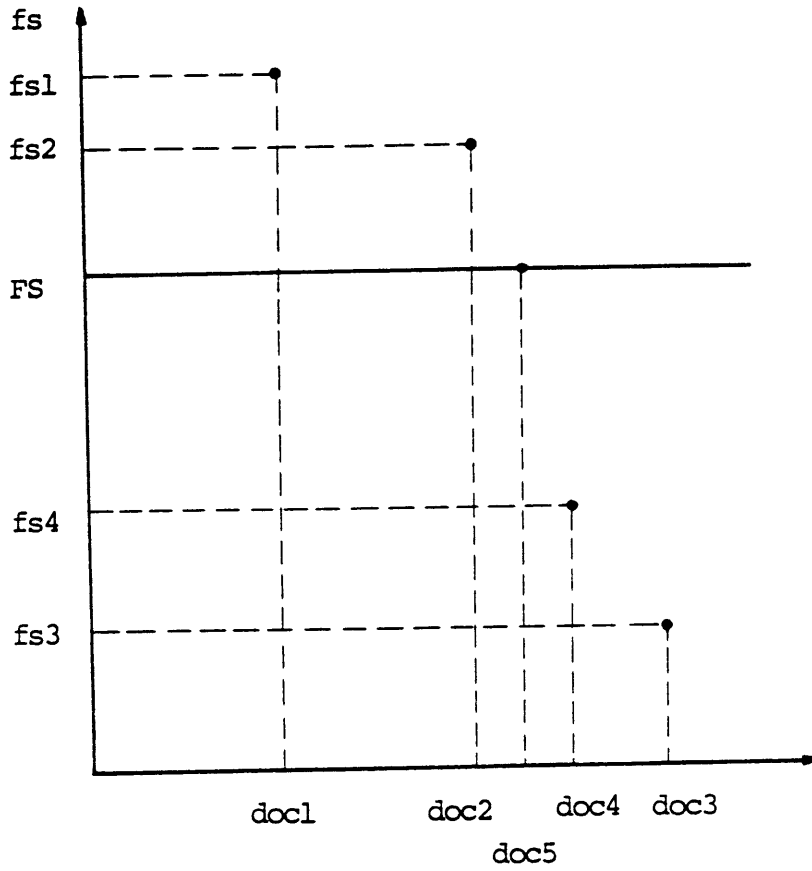


Figure 8 - Iteration scenario when specified fleet size is less than maximum fleet size

SEGMENT NO.: 101								U,L	1	2
11	12	13	14	15	16	17	18	19	20	21
TIME AXIS	DELETE	<u>ADD</u>	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 9 - Segment number input

ORIGIN: AAA								U,L	1	2
11	12	13	14	15	16	17	18	19	20	21
TIME AXIS	DELETE	<u>ADD</u>	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 10 - Origin station code name input

DESTINATION: BBB								U,L	1	2
11	12	13	14	15	16	17	18	19	20	21
TIME AXIS	DELETE	<u>ADD</u>	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 11 - Destination station code name input

VALUE: 100								U,L	1	2
11	12	13	14	15	16	17	18	19	20	21
TIME AXIS	DELETE	<u>ADD</u>	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 12 - Flight segment value input

FLYING TIME: 300								U,L	1	2
11	12	13	14	15	16	17	18	19	20	21
TIME AXIS	DELETE	<u>ADD</u>	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 13 - Flight segment duration input

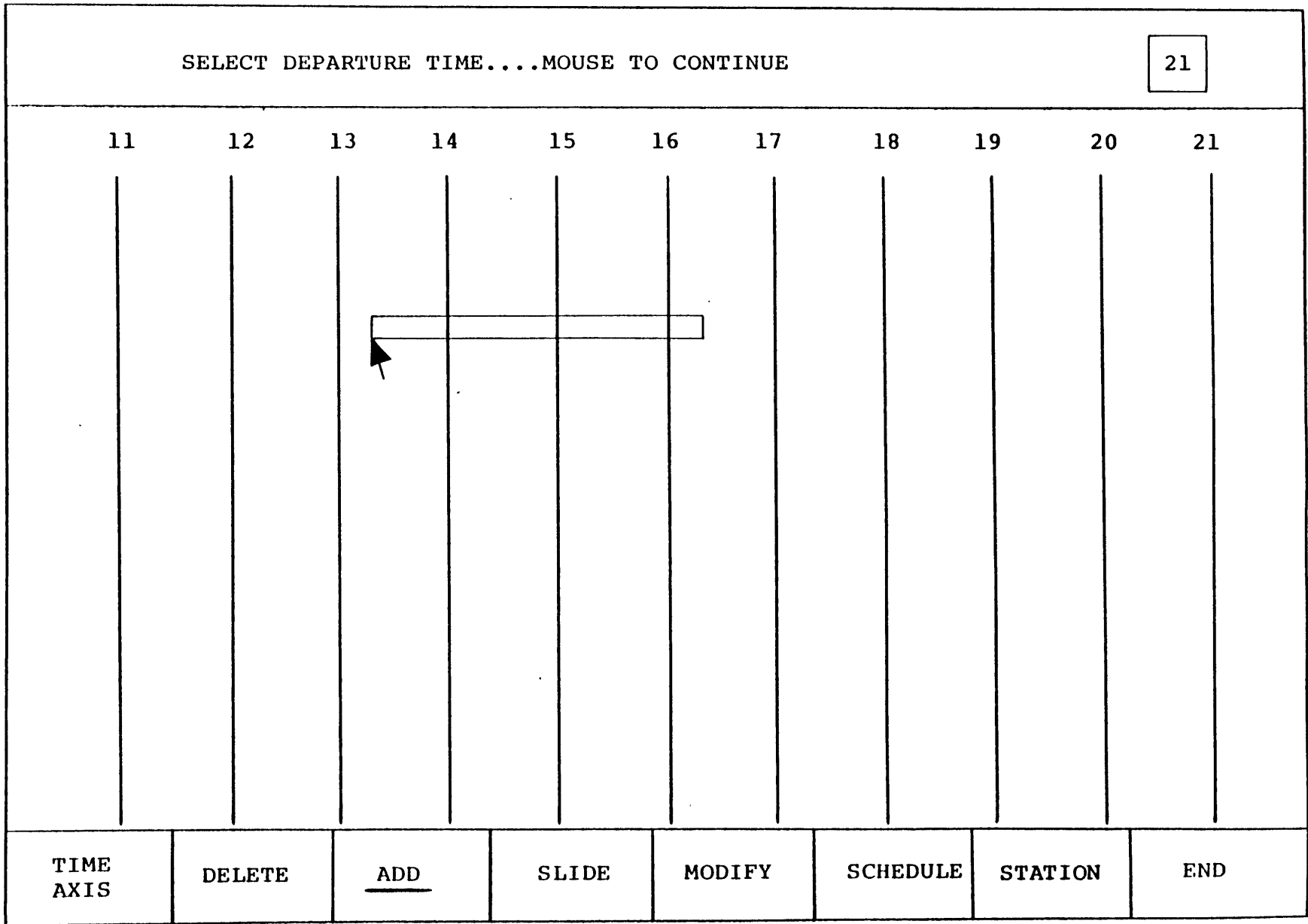


Figure 14 - The flight segment rectangle is moved to the appropriate position

OPTIMIZATION OPTIONS								U,L	1	2
11	12	13	14	15	16	17	18	19	20	21
			AAA	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> 0100 101 </div>			BBB			
			00			00				
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 15 - Flight segment 101 has been scheduled

SELECT FLIGHT DATA ELEMENT TO BE CHANGED

SEGMENT NUMBER = 101

ORIGIN = AAA

DESTINATION = BBB

FLYING TIME = 0300

ARRIVAL TIME = 1710

VALUE = 0100

TIME
AXIS

DELETE

ADD

SLIDE

MODIFY

SCHEDULE

STATION

END

Figure 16 - Flight segment characteristics display

NEW VALUE: 200							
SEGMENT NUMBER = 101							
ORIGIN = AAA							
DESTINATION = BBB							
FLYING TIME = 0300							
ARRIVAL TIME = 1710							
VALUE = 100							
TIME AXIS	DELETE	ADD	SLIDE	<u>MODIFY</u>	SCHEDULE	STATION	END

Figure 17 - Flight segment 101's value is increased

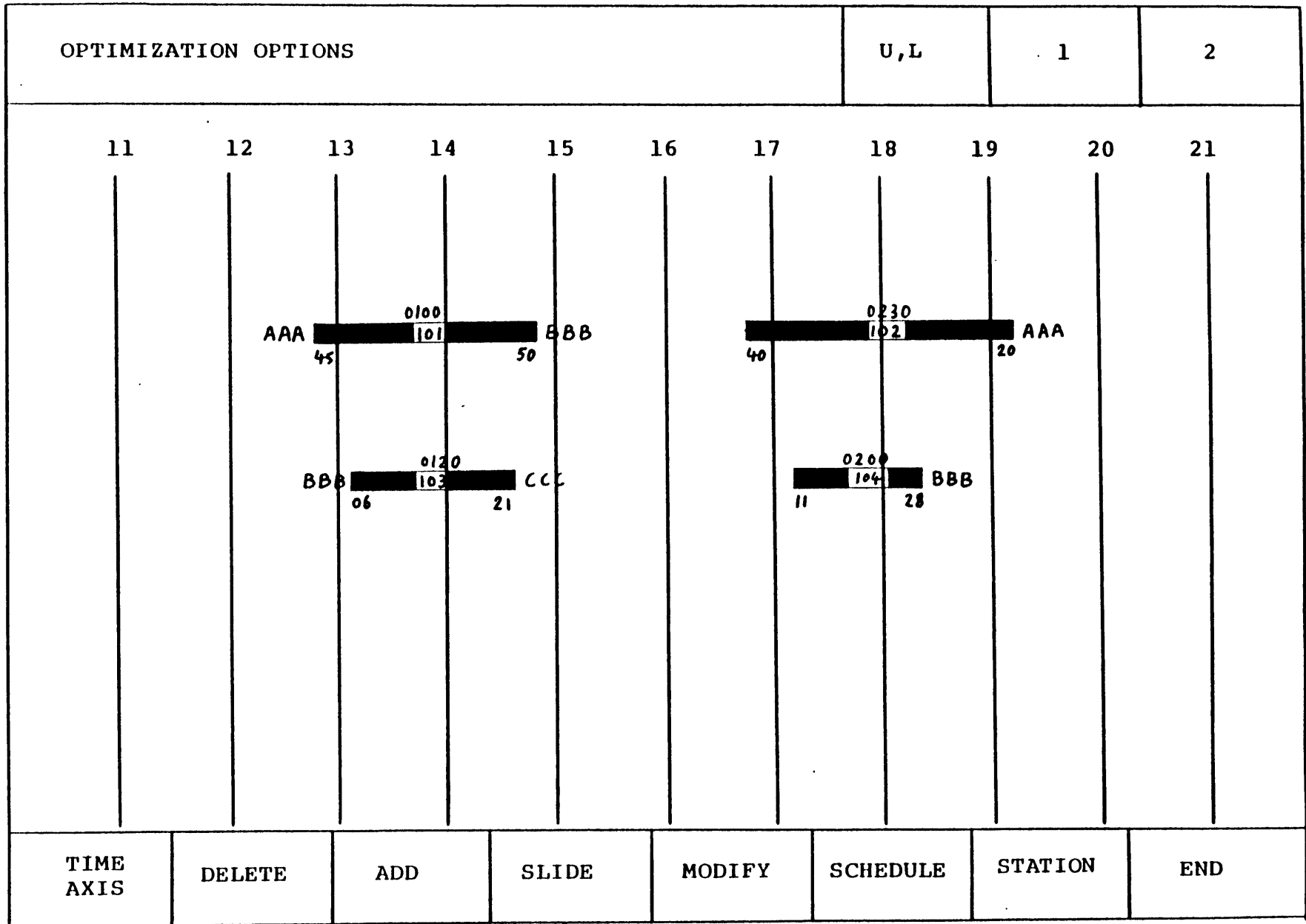


Figure 18 - A schedule plan consisting of four flight segments

SELECT SEGMENT TO BE MOVED....MOUSE DURING FLYING TIME

43

11 12 13 14 15 16 17 18 19 20 21

AAA ⁰¹⁰⁰ 101 BBB
45 50

⁰²³⁰ 102 AAA
40 20

BBB ⁰¹²⁰ 103 CCC
06 21

⁰²⁰⁰ 104 BBB
11 28



TIME
AXIS

DELETE

ADD

SLIDE

MODIFY

SCHEDULE

STATION

END

59

Figure 19 - Flight segment 104 is about to be selected for rescheduling

SELECT NEW DEPARTURE TIME FOR FLIGHT SEGMENT

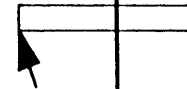
22

11 12 13 14 15 16 17 18 19 20 21

AAA 0100
45 101 50 BBB

0230
40 102 20 AAA

0120
BBB 06 103 21 CCC



TIME
AXIS

DELETE

ADD

SLIDE

MODIFY

SCHEDULE

STATION

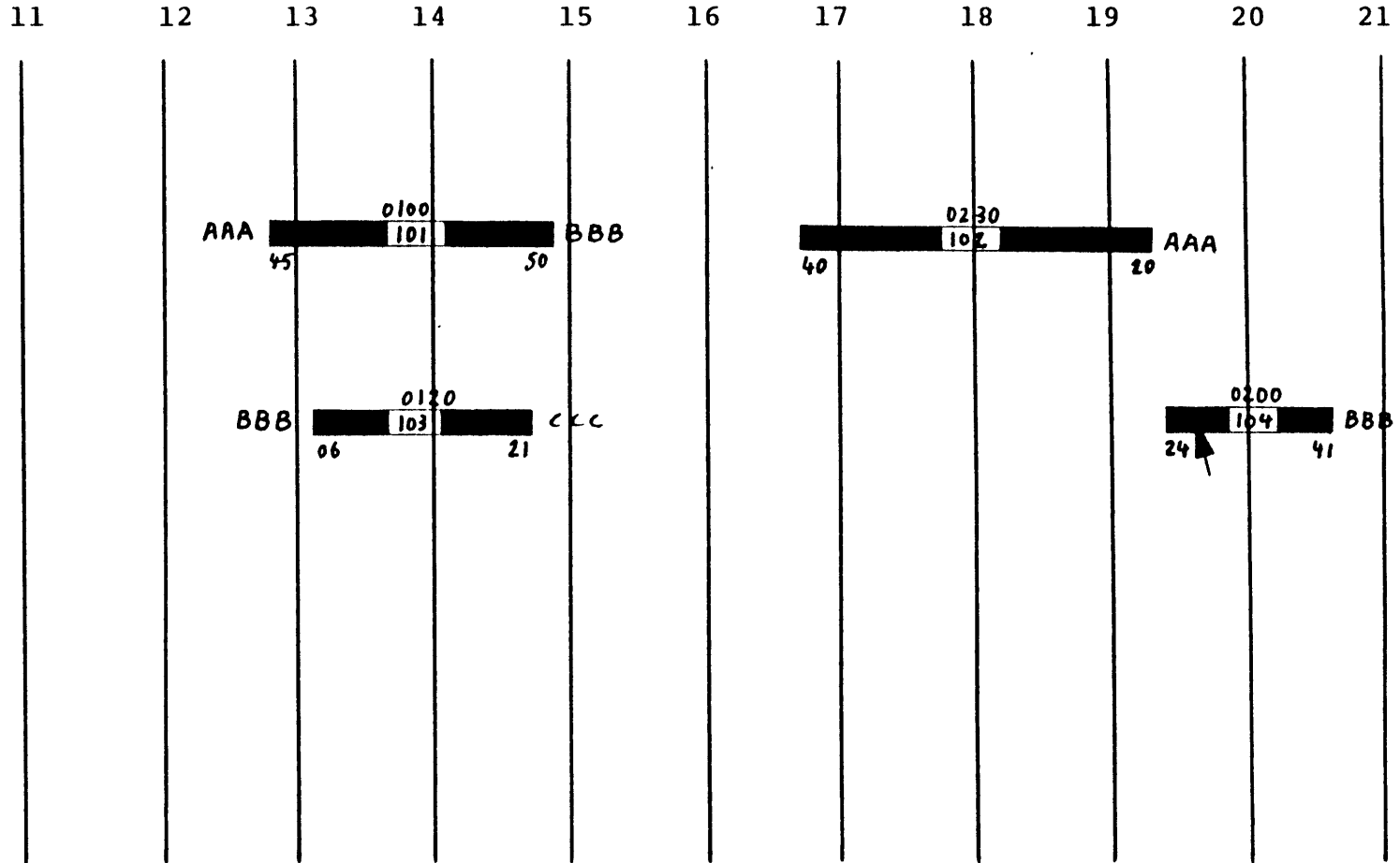
END

89

Figure 20 - Flight segment 104 is being rescheduled

SELECT SEGMENT TO BE DELETED....MOUSE DURING FLYING TIME

38



TIME
AXIS

DELETE

ADD

SLIDE

MODIFY

SCHEDULE

STATION

END

Figure 21 - Flight segment 104 is about to be deleted

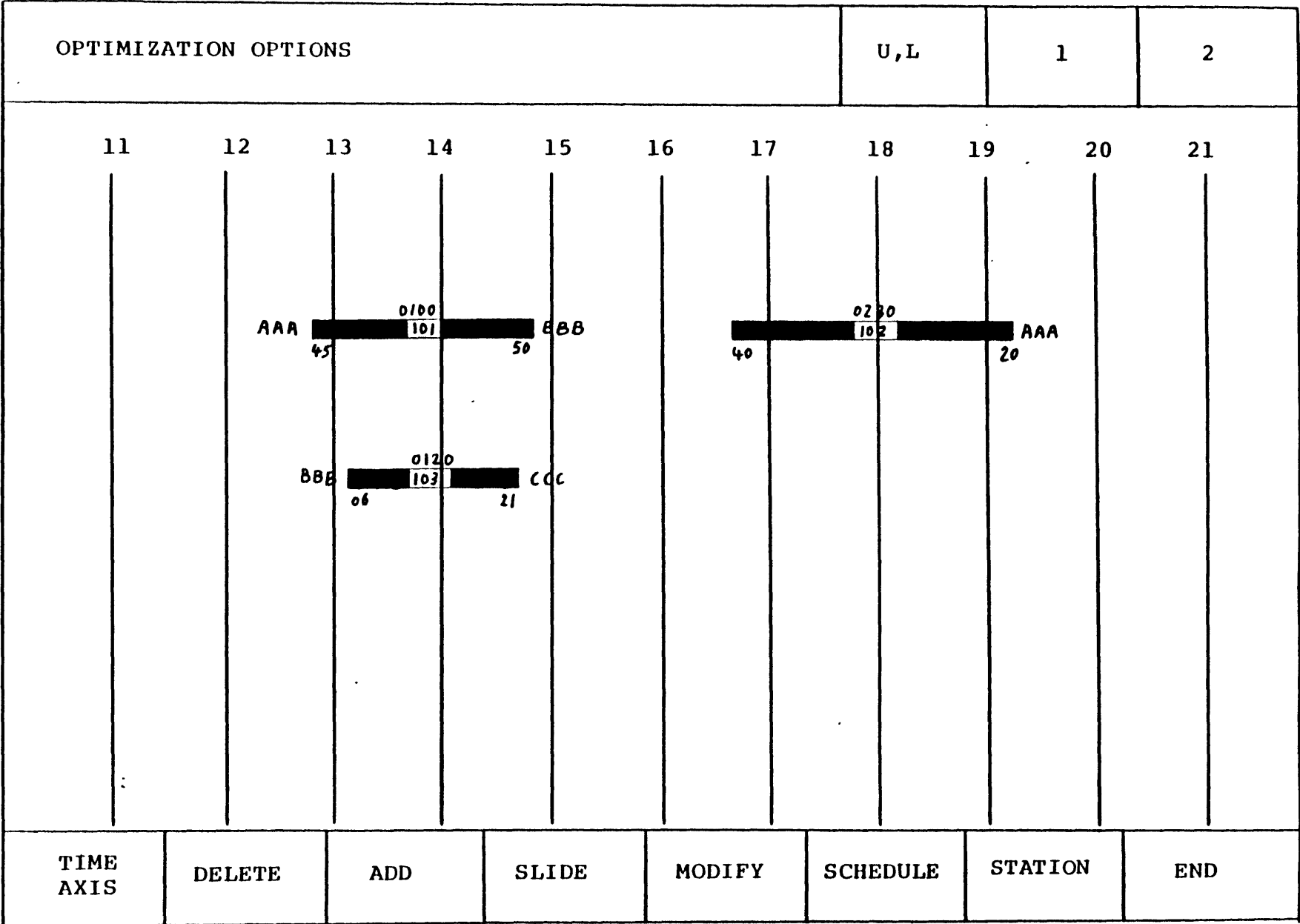
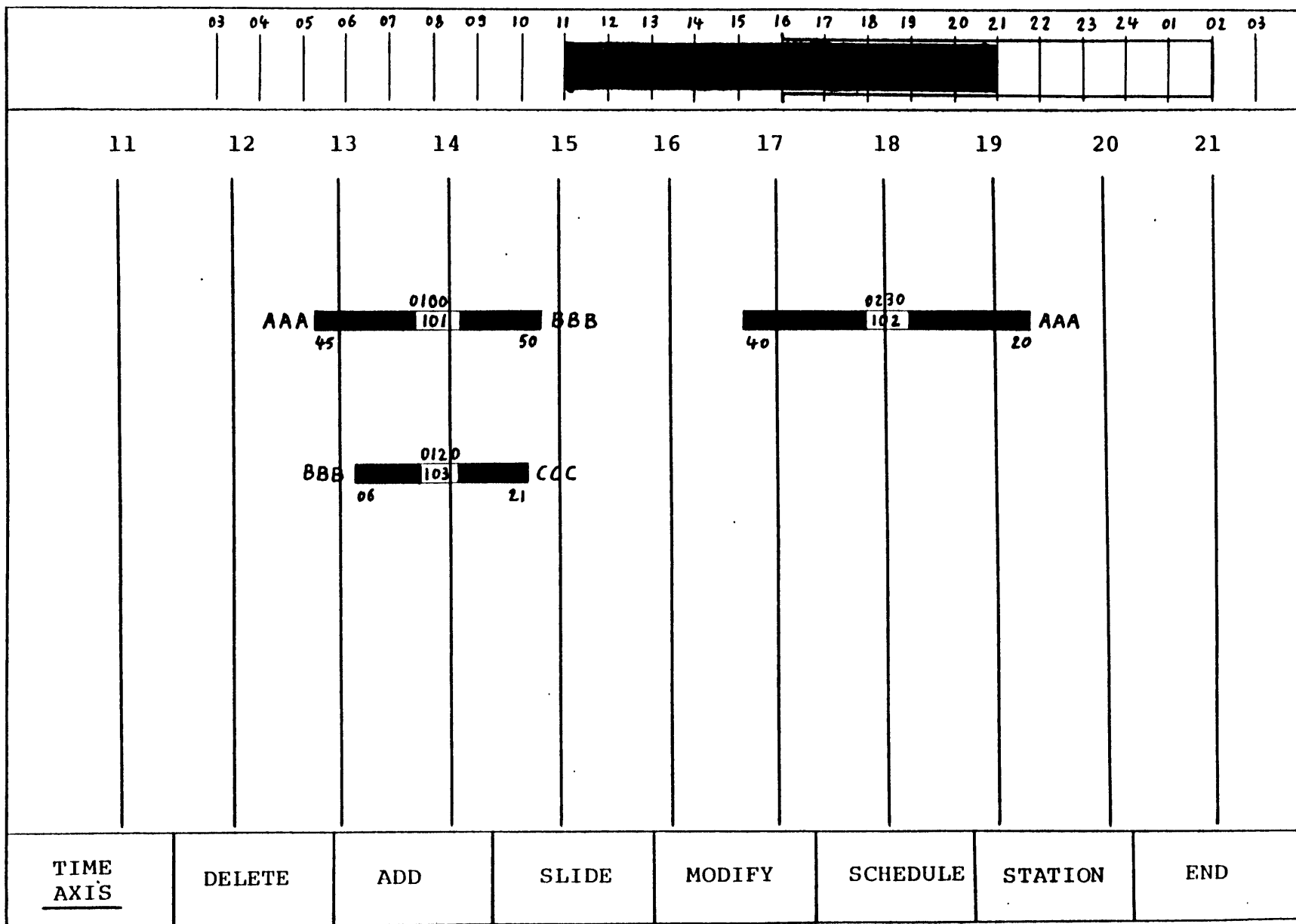


Figure 22 - Flight segment 104 has been deleted



63

Figure 23 - A new time interval is selected

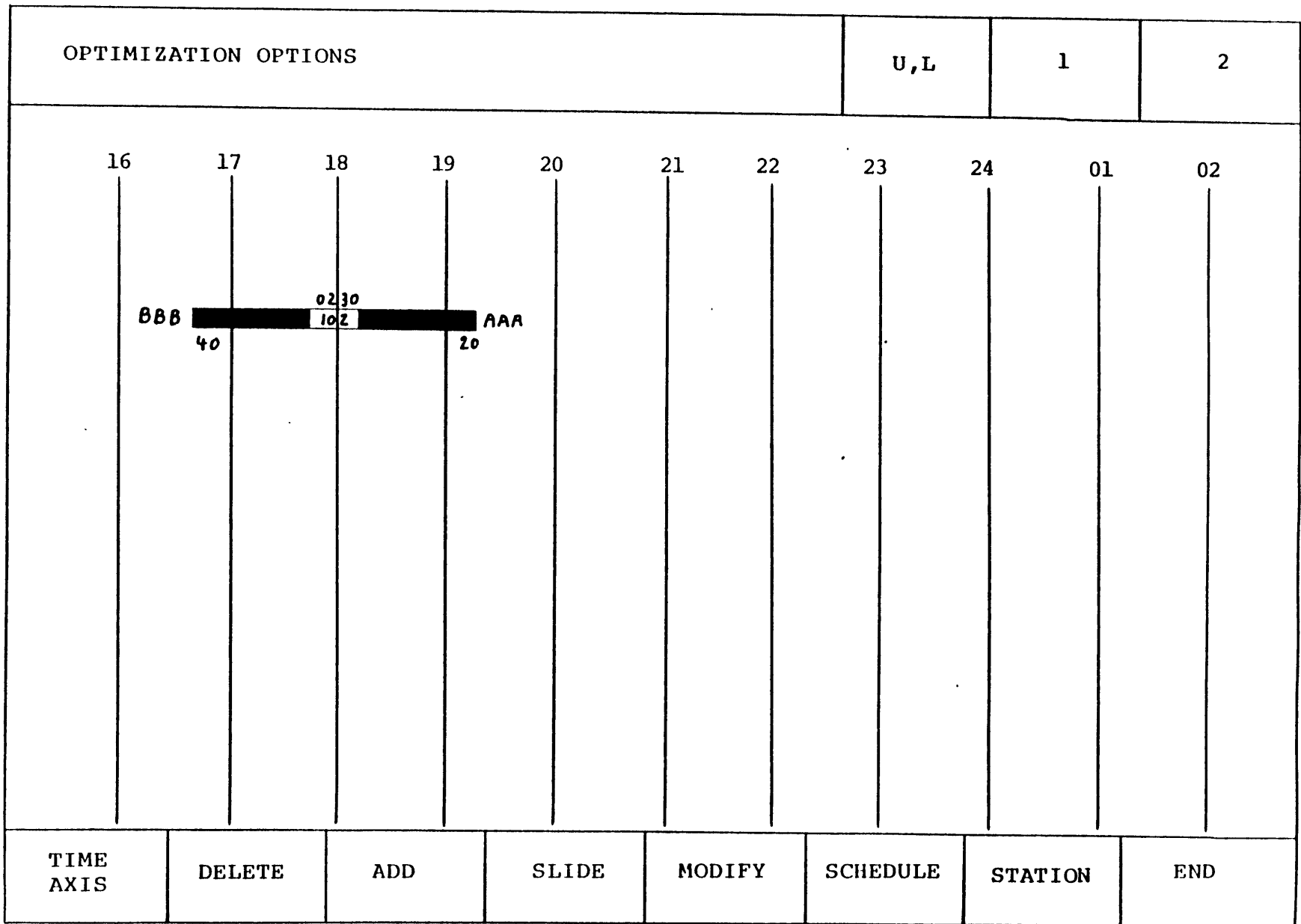


Figure 24 - The visible portion of the schedule chart
has been changed

SCHEDULE DATABASE							
<u>FLT#</u>	<u>ORIG</u>	<u>DEST</u>	<u>DEPT</u>	<u>ARRV</u>			
101	AAA	BBB	1145	1450			
102	BBB	AAA	1640	1920			
103	BBB	CCC	1306	1421			

TIME AXIS	DELETE	ADD	SLIDE	MODIFY	<u>SCHEDULE</u>	STATION	END
--------------	--------	-----	-------	--------	-----------------	---------	-----

Figure 25 - The schedule plan of Figure 23 is tabulated

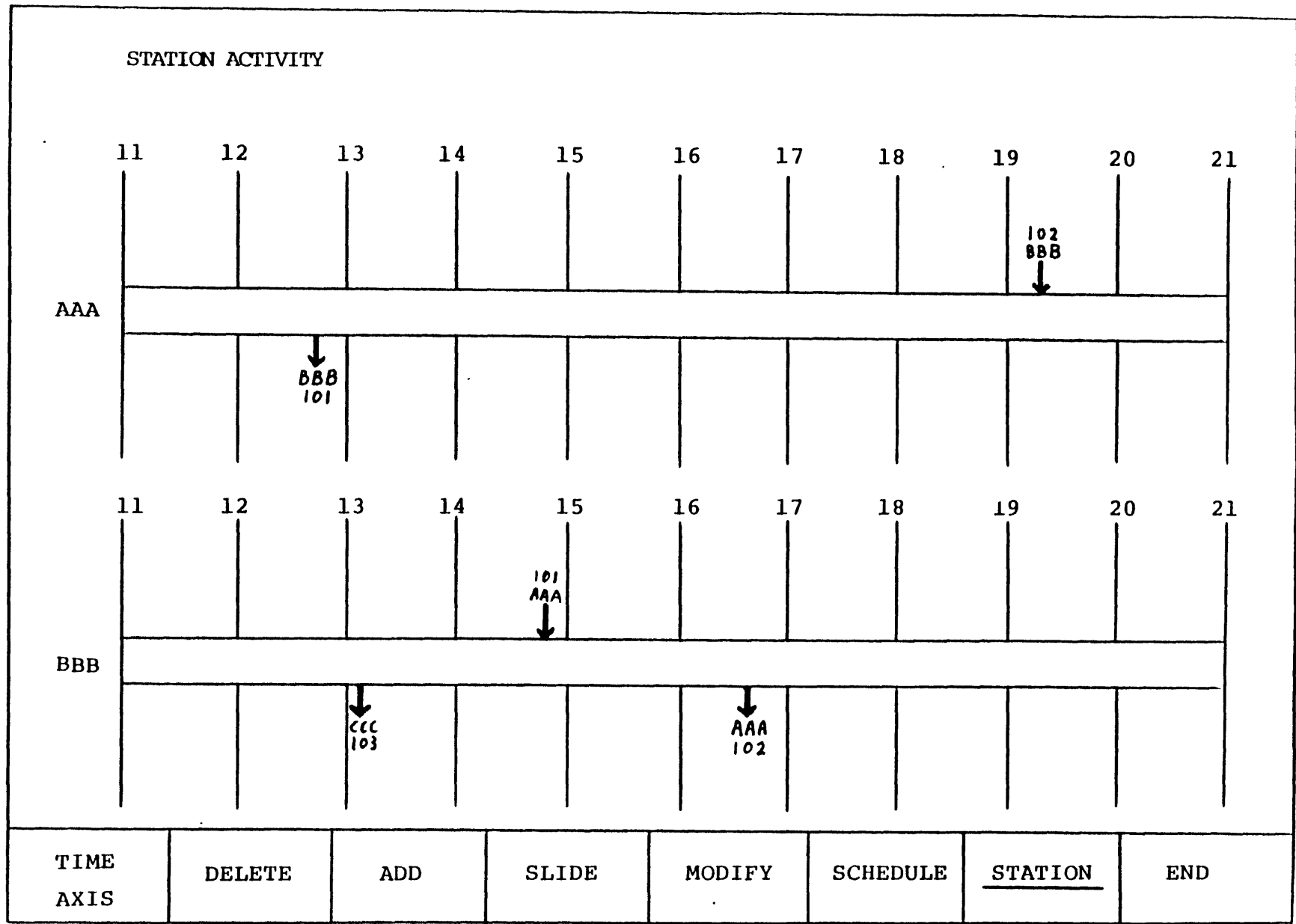


Figure 26 - Station display for stations AAA and BBB

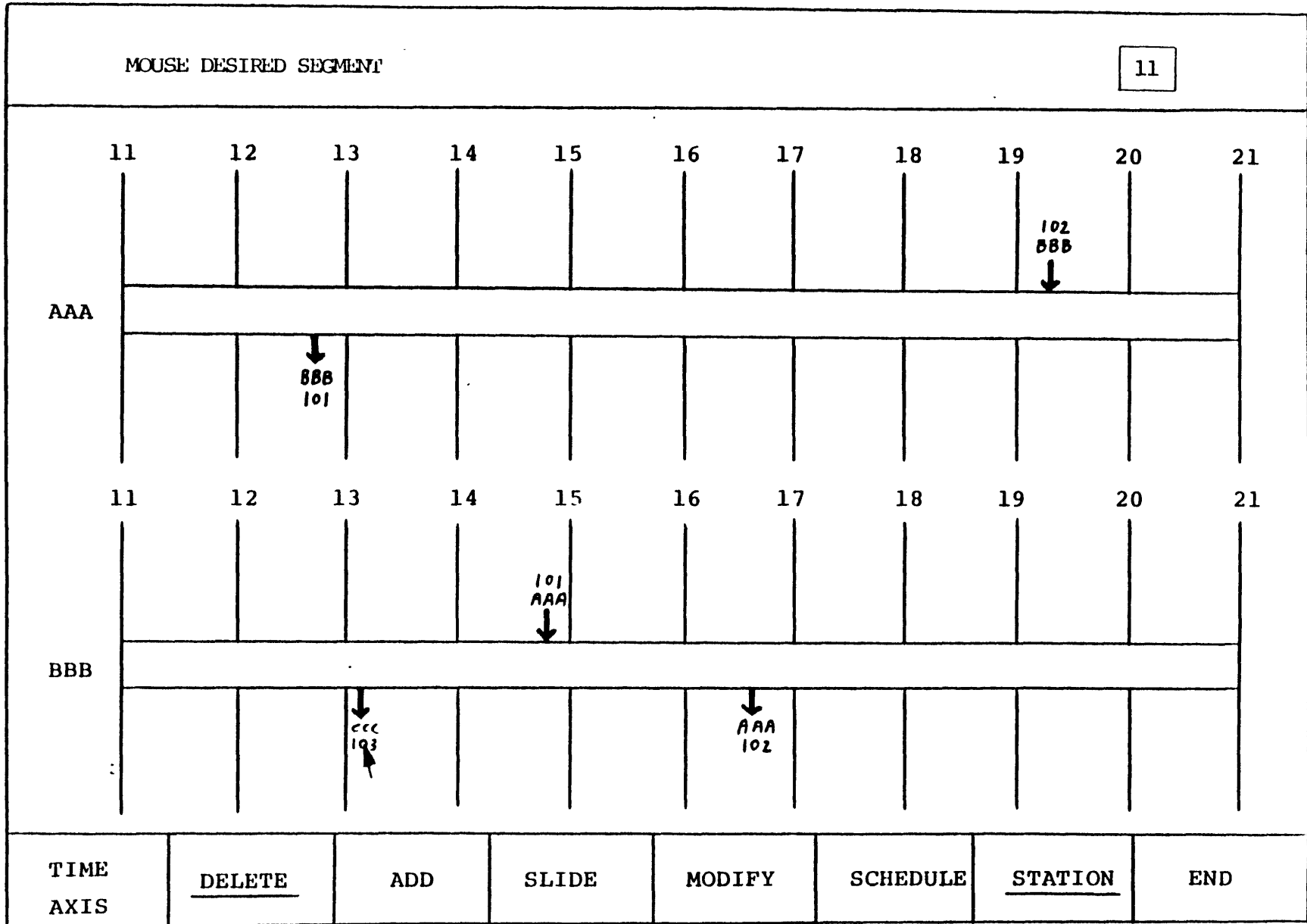


Figure 27 - Flight segment 103 will be deleted

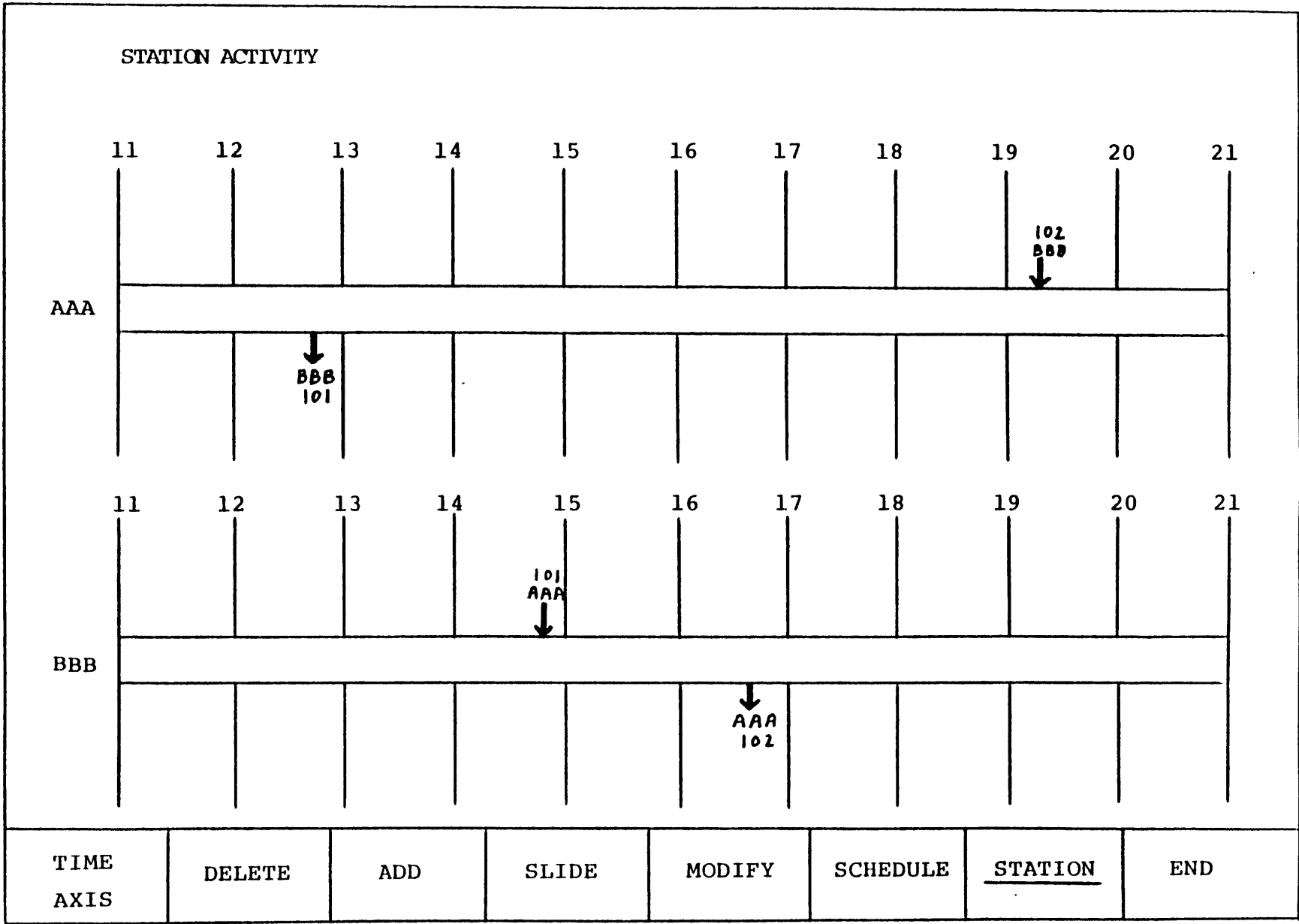


Figure 28 - Flight segment 103 has been deleted

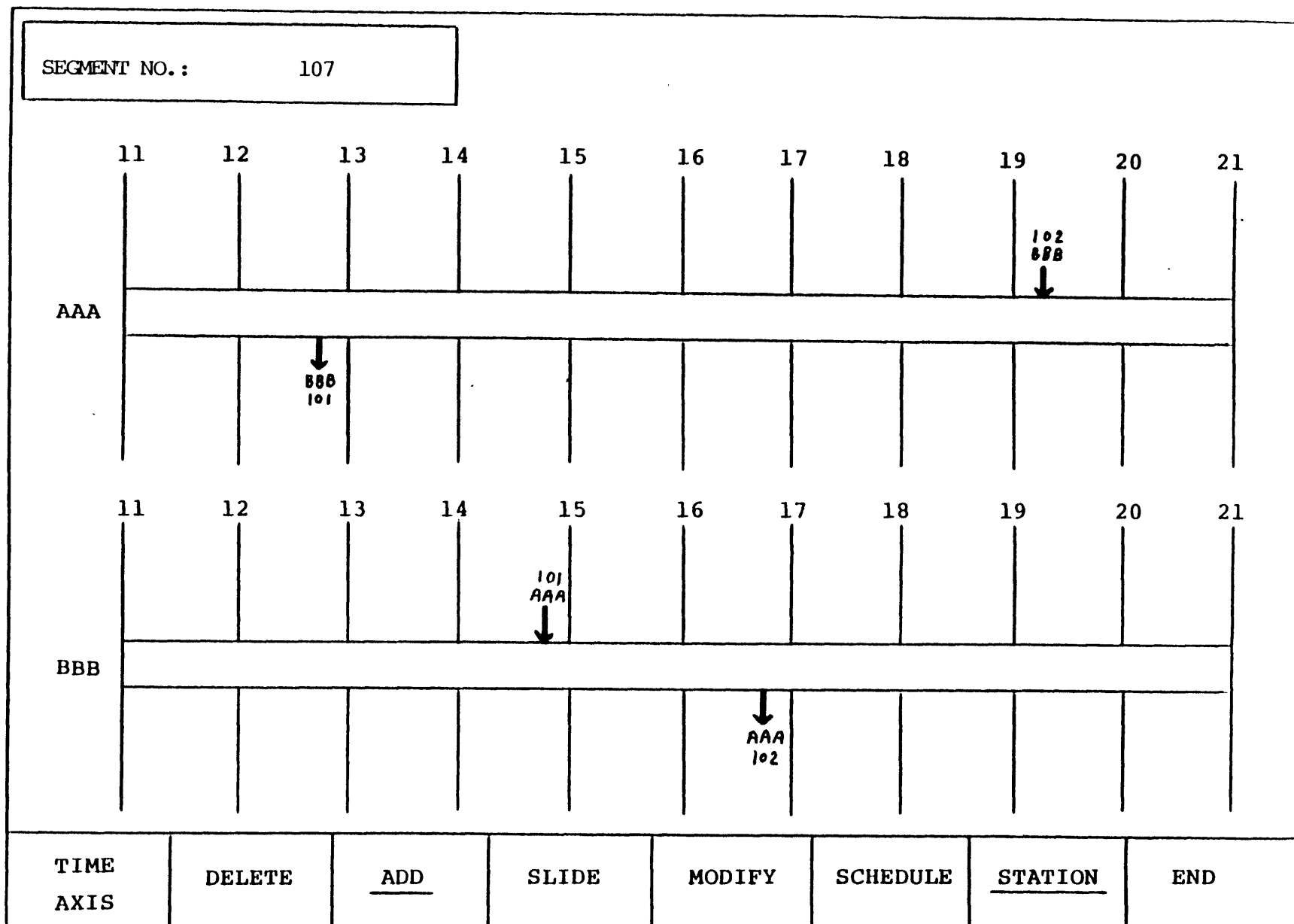


Figure 29 - Flight segment 107 is being added to
the schedule plan

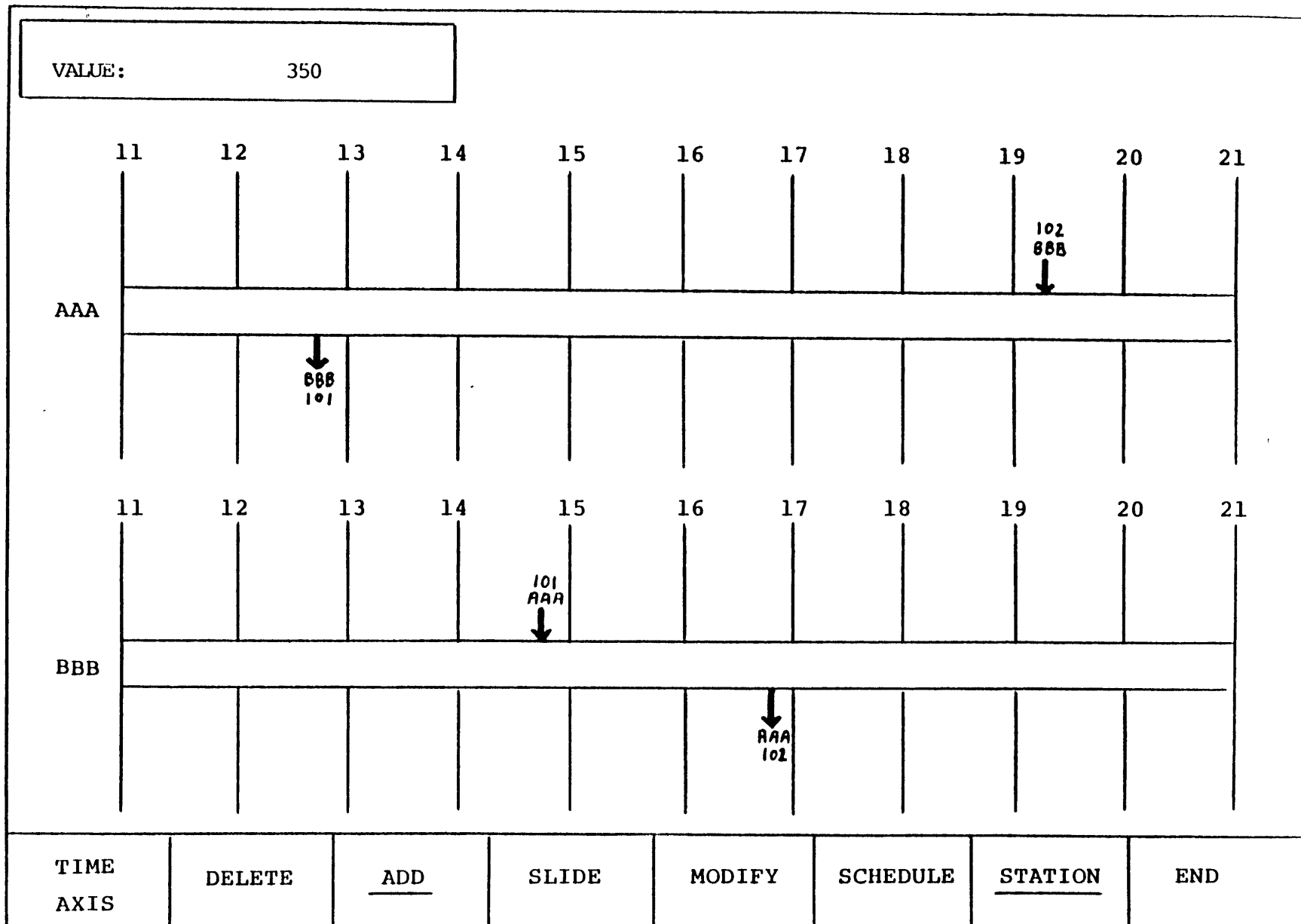


Figure 30 - The value of flight segment 107 is 350

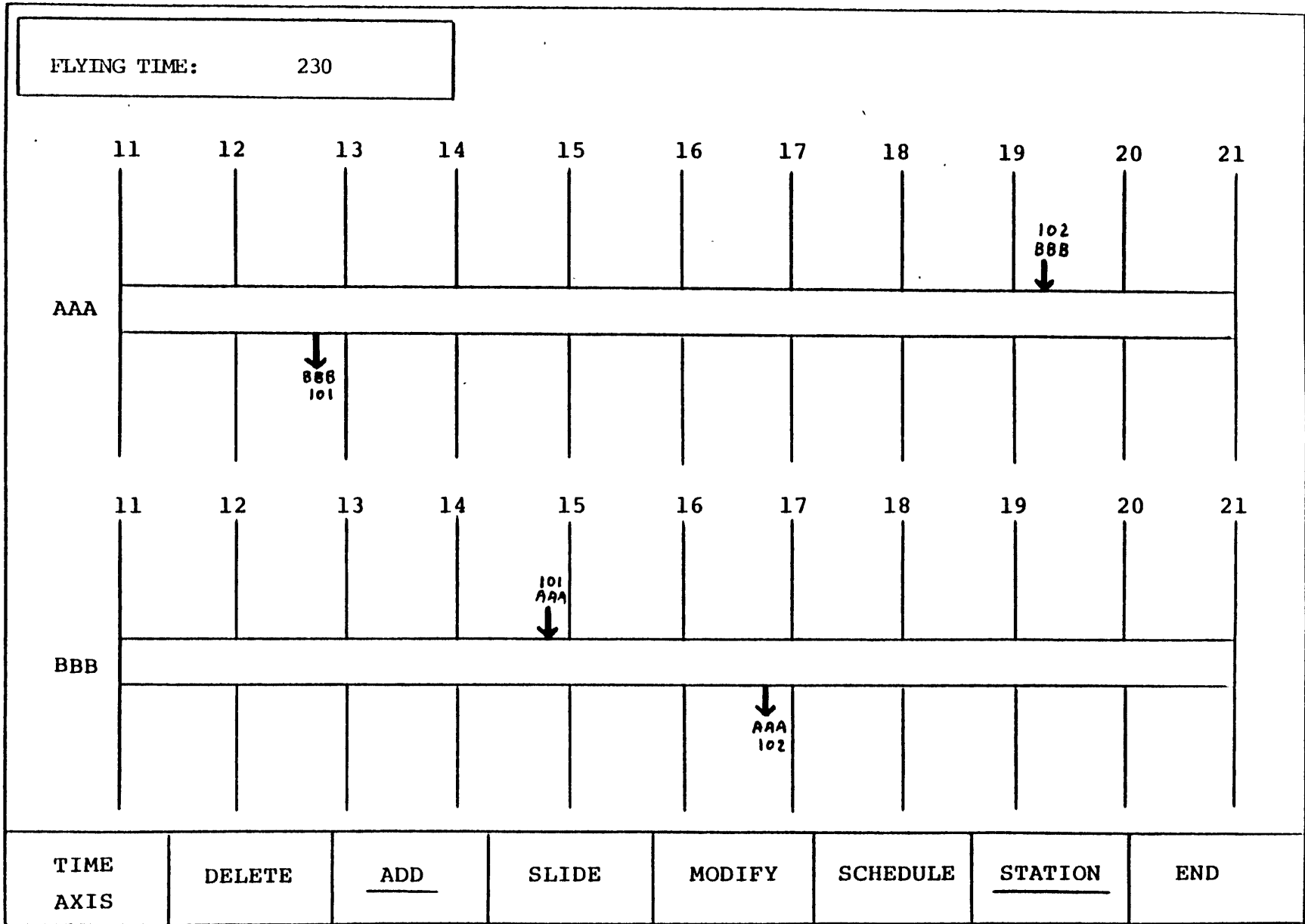


Figure 31 - The duration of the flight segment is 2.5 hrs

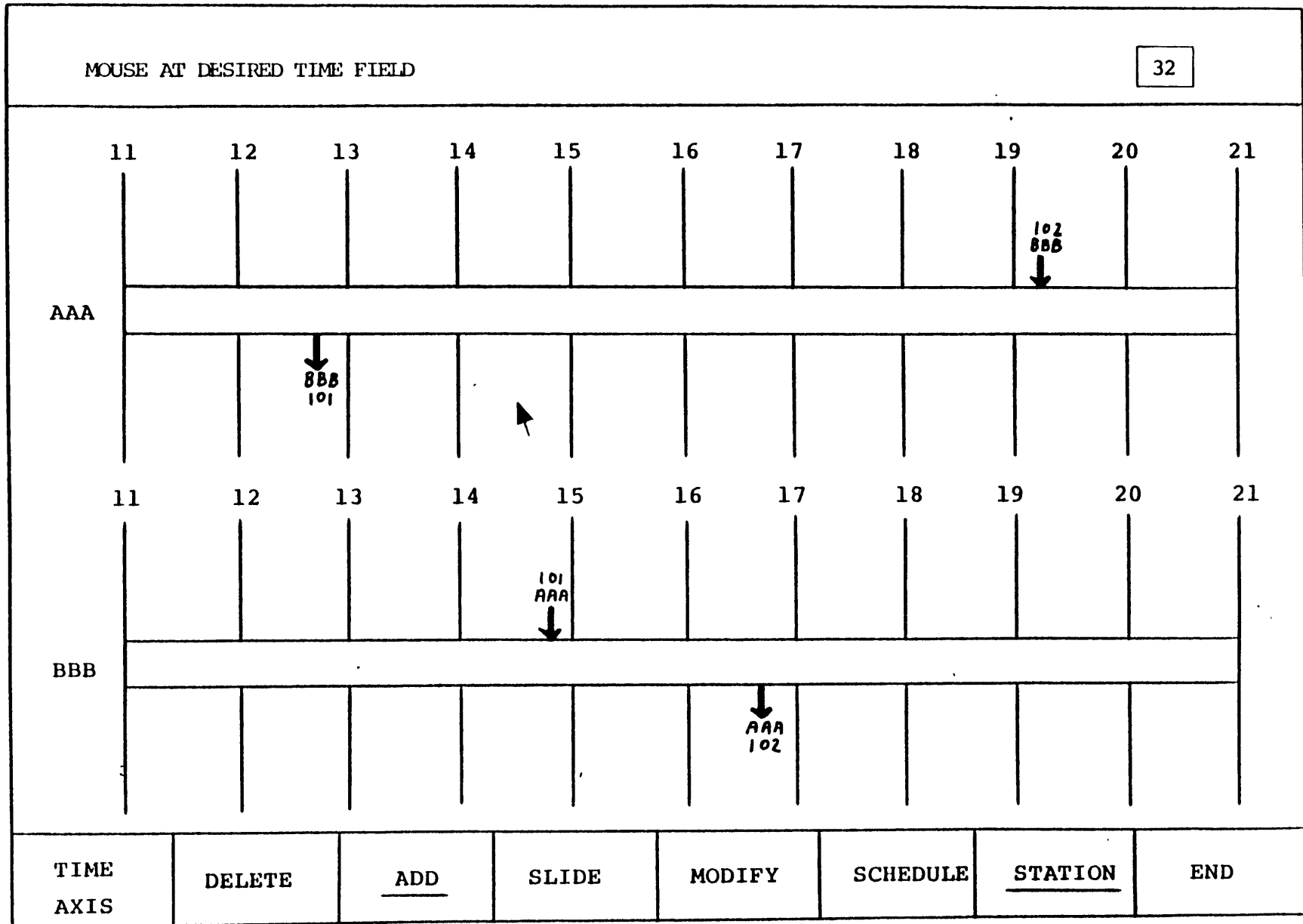


Figure 32 - Flight segment 107's departure time is selected

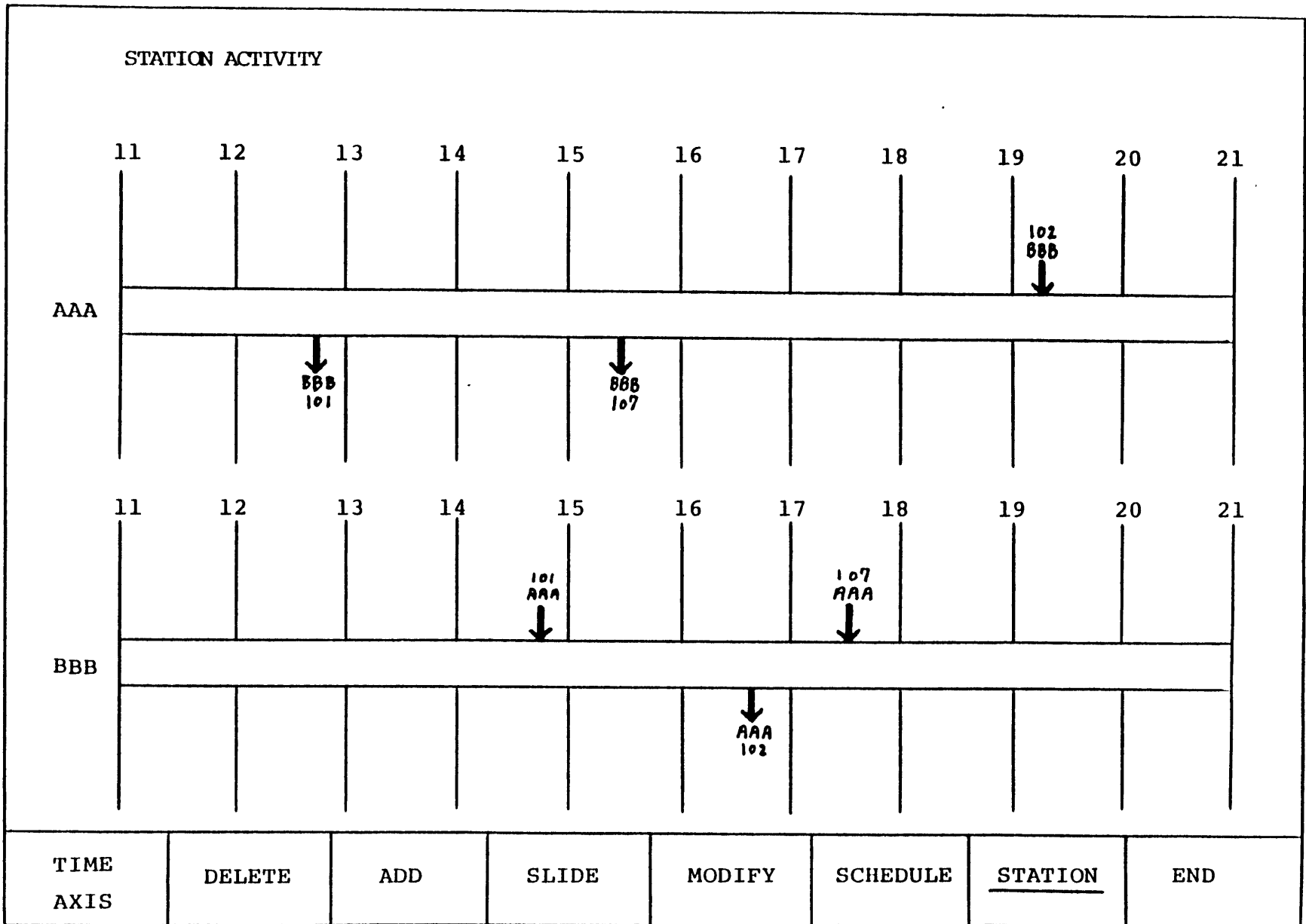


Figure 33 - Flight segment 107 has been scheduled

STATION ACTIVITY										
11	12	13	14	15	16	17	18	19	20	21
AAA				1						
					2					
BBB					3					
					4					
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	<u>STATION</u>	END			

Figure 34 - The "time fields" for the selection of arrival or departure times

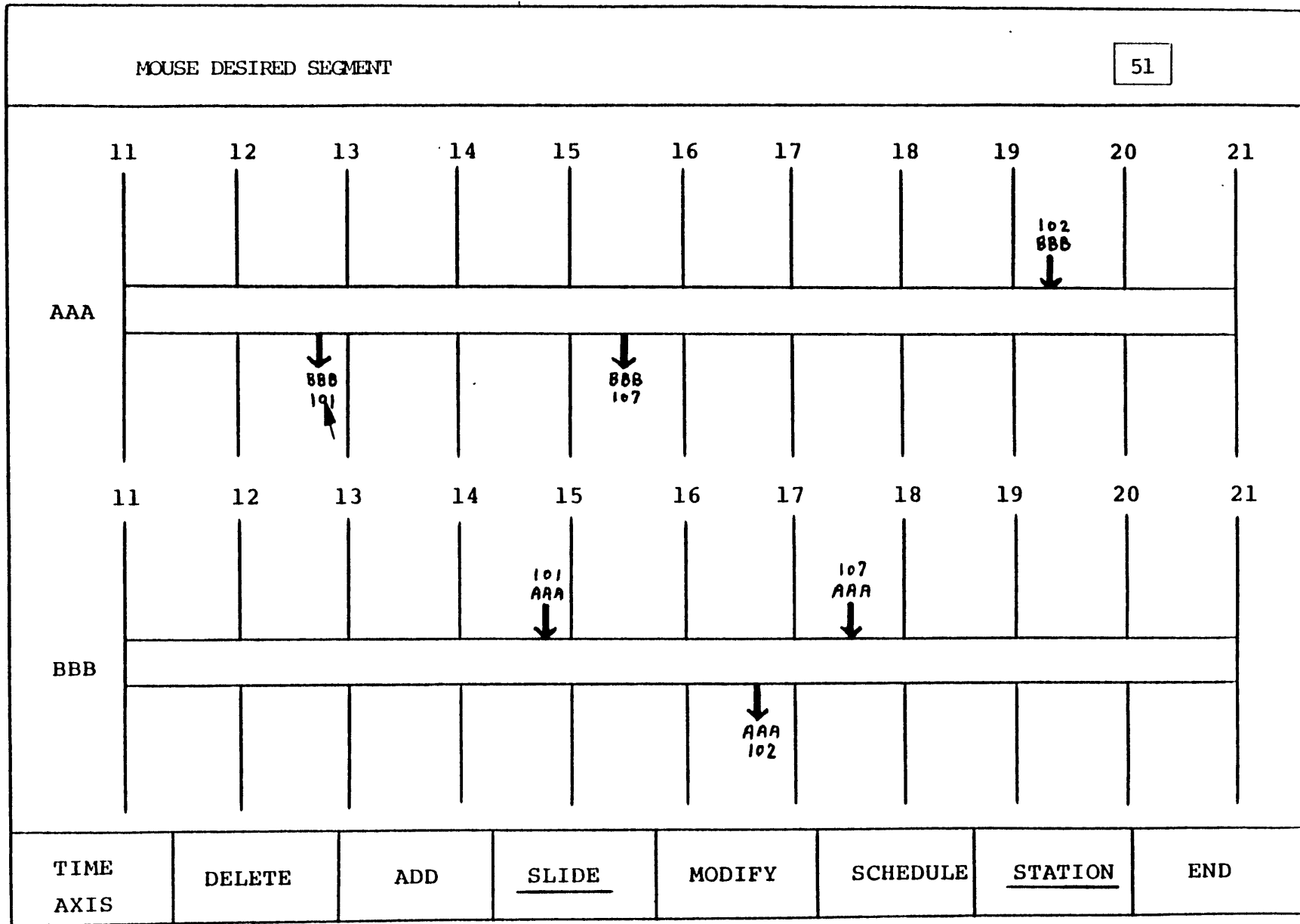


Figure 35 - Flight segment 101 is selected for rescheduling

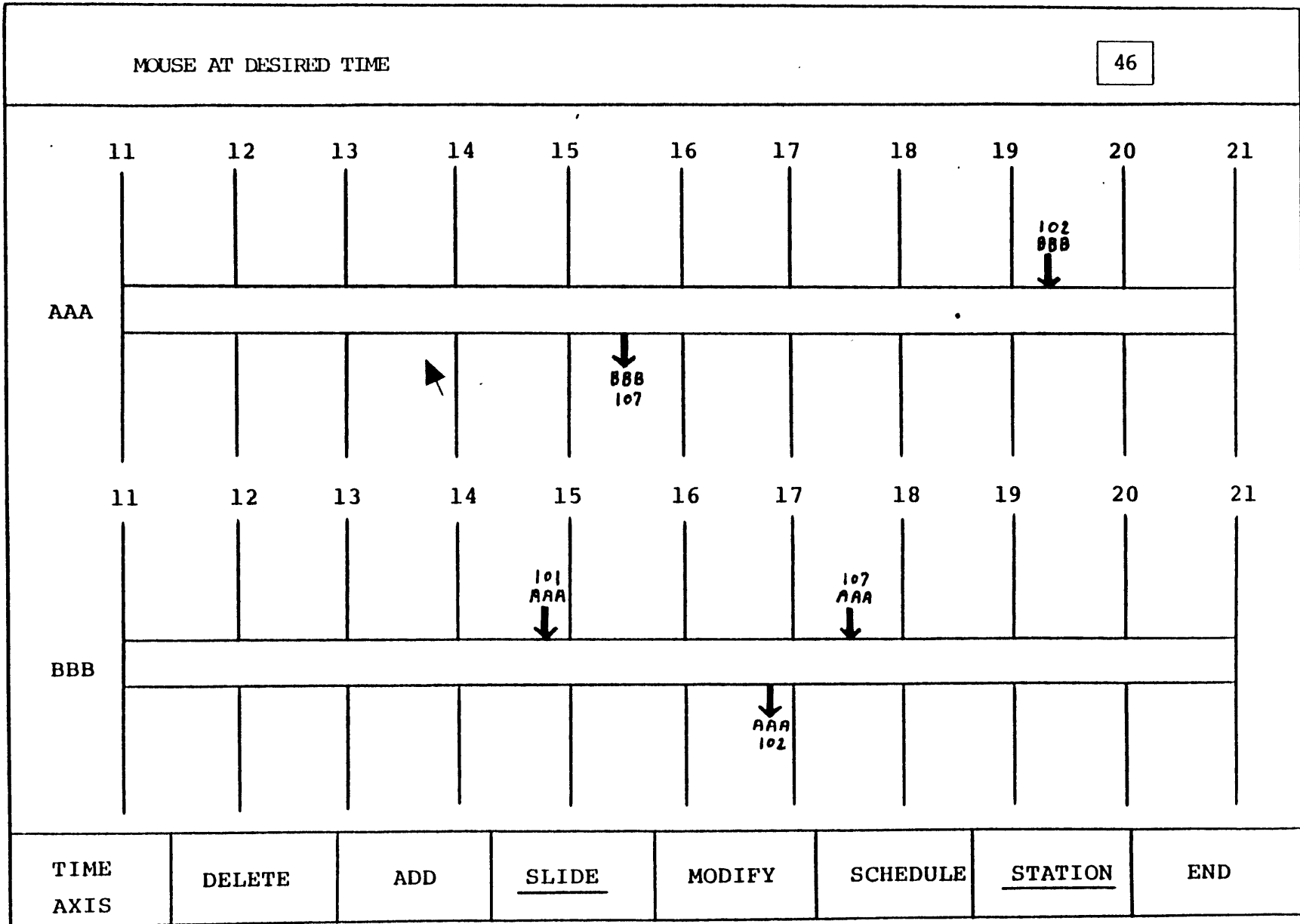


Figure 36 - The arrow is moved to the new departure time

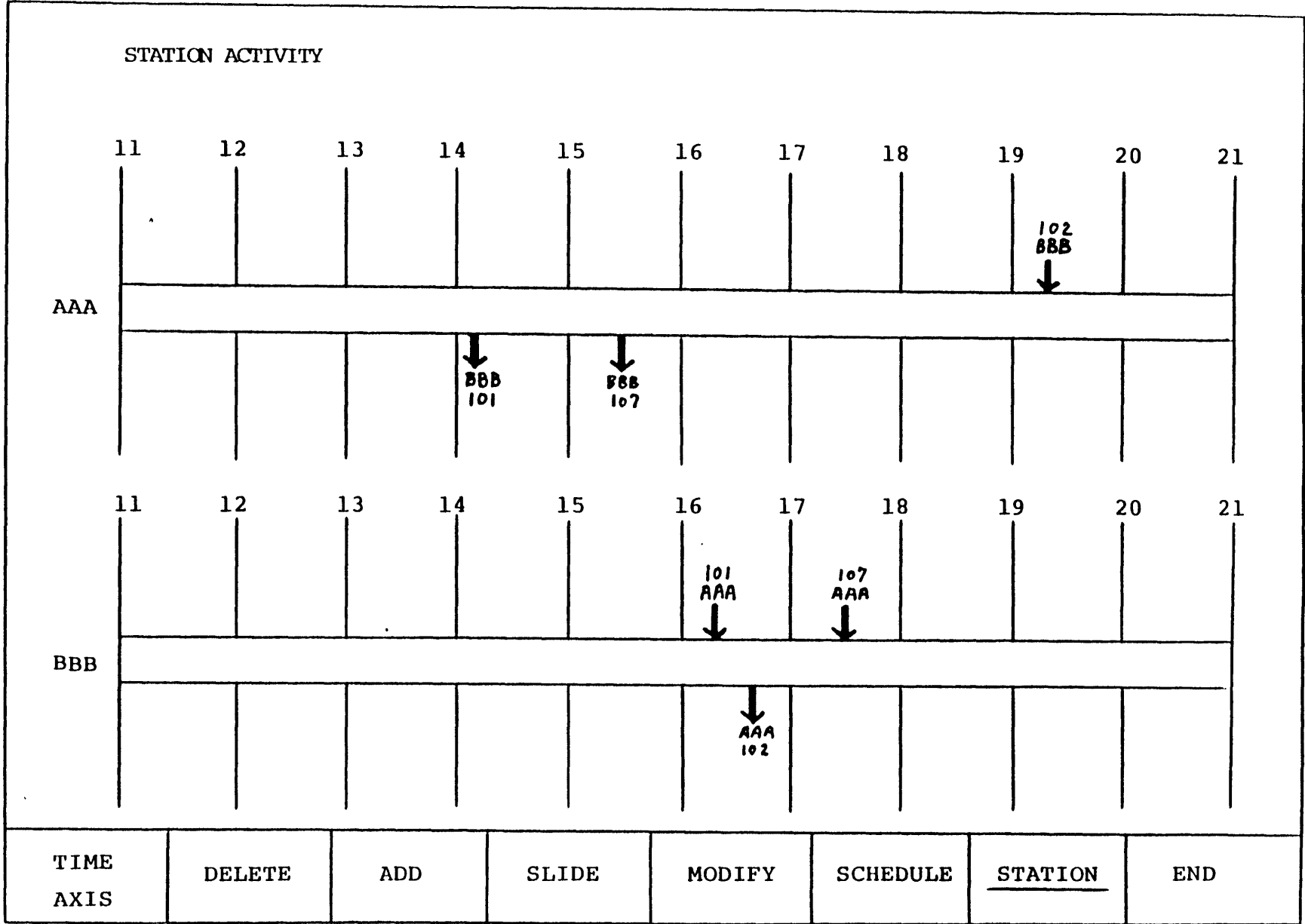


Figure 37 - Flight segment 101 has been rescheduled

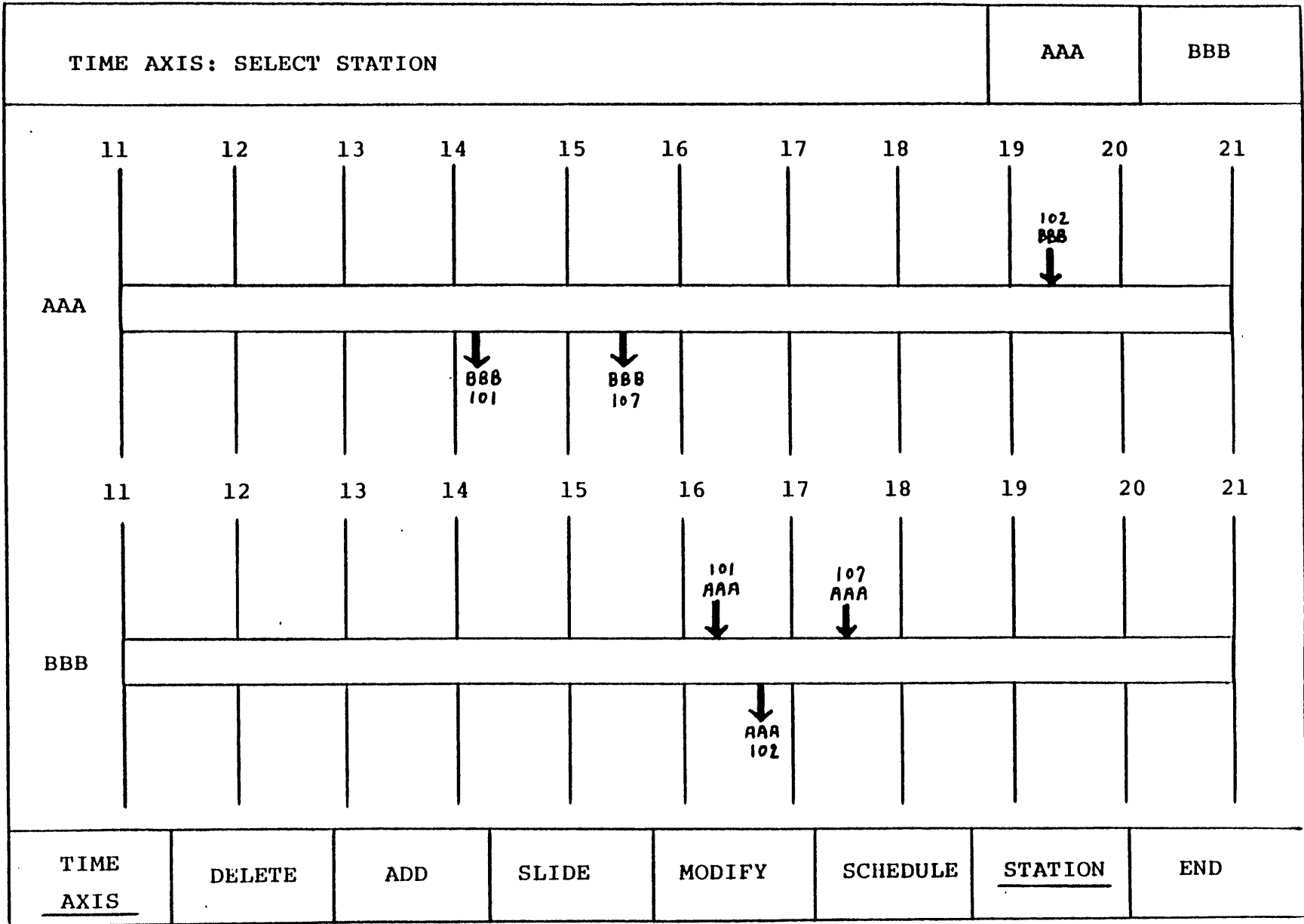


Figure 38 - The time axis of either AAA or BBB
can be translated

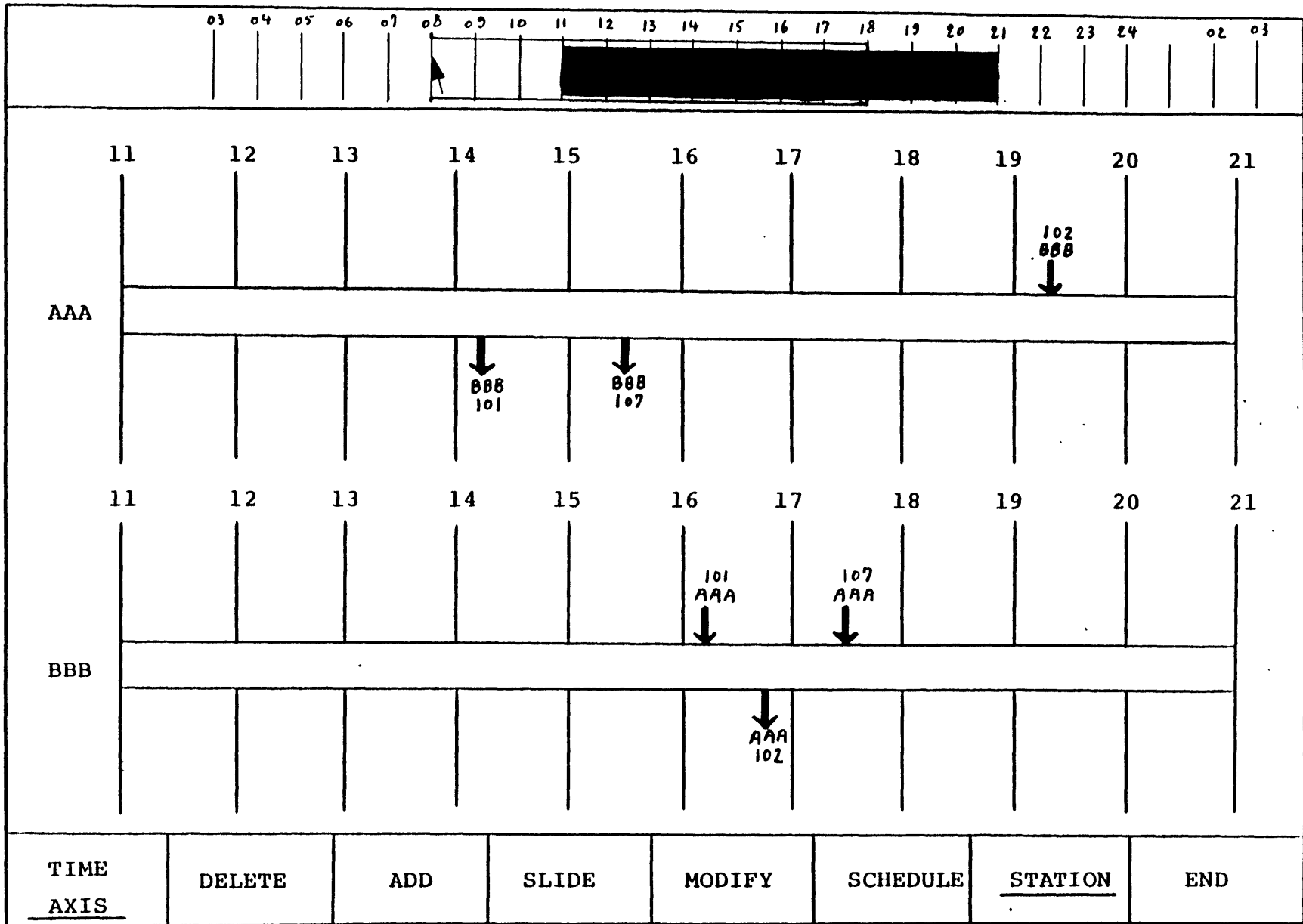


Figure 39 - BBB's visible time axis interval will be from 0800 to 1800

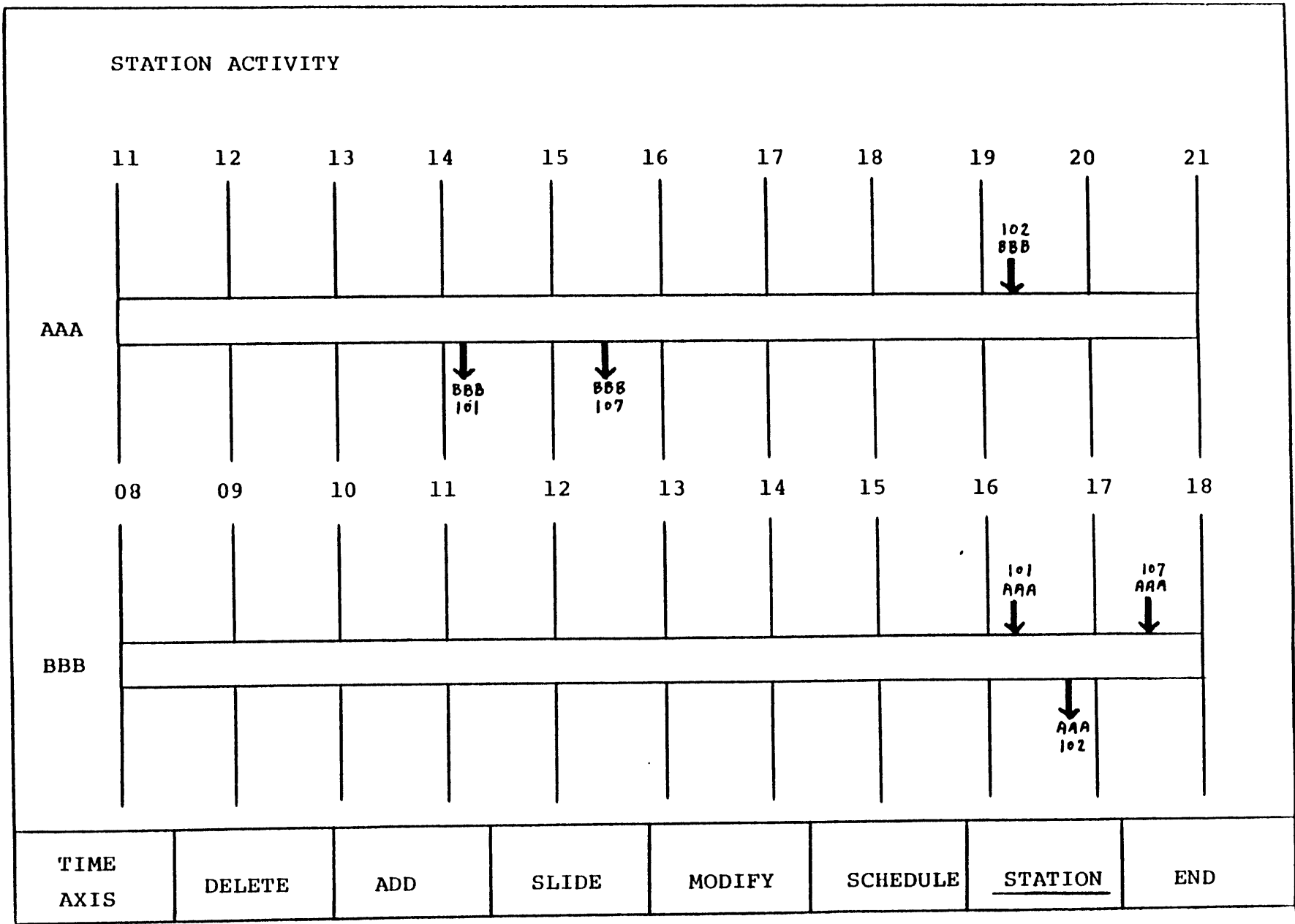


Figure 40 - BBB's time axis has been translated

OPTIMIZATION OPTIONS							U,L	1	2	
11	12	13	14	15	16	17	18	19	20	21
			AAA 12	0100 101	BBB 17	40	0230 102	20	AAA	
				AAA 30	0350 107	BBB 30				
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END			

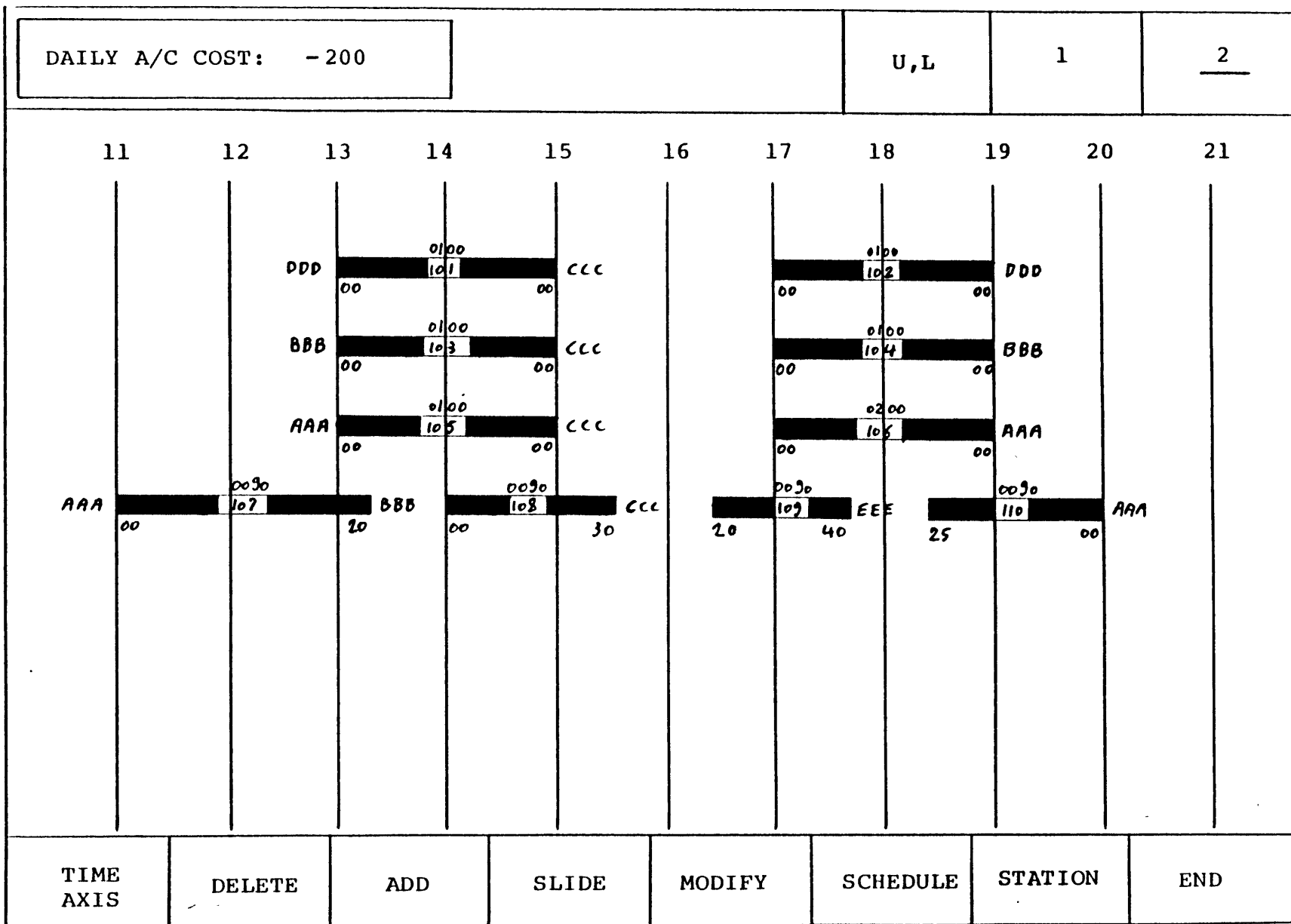
Figure 41 - Schedule plan resulting from previous
flight segment manipulations

OPTIMIZATION OPTIONS				U,L	<u>1</u>	2	
<p>fleet size: 004</p> <p>overnight stations: DDD(01), BBB(01), AAA(02)</p> <p style="text-align: right;">mouse to continue</p>							
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END

Figure 43 - Answer to the first optimization option

NO. OF A/C: 1						U,L	1	2					
11	12	13	14	15	16	17	18	19	20	21			
AAA 00	0090 107	BBB 00	0100 101	CCC 00		0100 102	DDD 00	0100 103	BBB 00	0100 104	AAA 00	0200 106	AAA 00
			0100 105	CCC 00	0090 108	CCC 30	0090 105	EEE 40	0090 10	AAA 00			
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END						

Figure 44 - Fleet size input for the fleet routing option



85

Figure 45 - Generic daily ownership cost (doc) input
for the fleet routing option

OPTIMIZATION OPTIONS						U,L	1	<u>2</u>
fleet size: 001 fleet net take: 0160 max doc: -360 max fleet size: 001 overnight stations: AAA(01)								
grey flights are flown						mouse to continue		
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END	

Figure 46 - Answer to the second optimization option
with fs=1 and doc=-200

OPTIMIZATION OPTIONS					U,L	1	<u>2</u>
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END

Figure 47 - Flight of aircraft 001

OPTIMIZATION OPTIONS				U,L	1	<u>2</u>
<p>fleet size: 004</p> <p>fleet net take: -0940</p> <p>max doc: -0265</p> <p>overnight stations: . DDD(01), BBB(01), AAA(02)</p> <p>grey flights are flown mouse to continue</p>						
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION END

Figure 48 - Answer to the second optimization option
with fs=4 and doc=-500

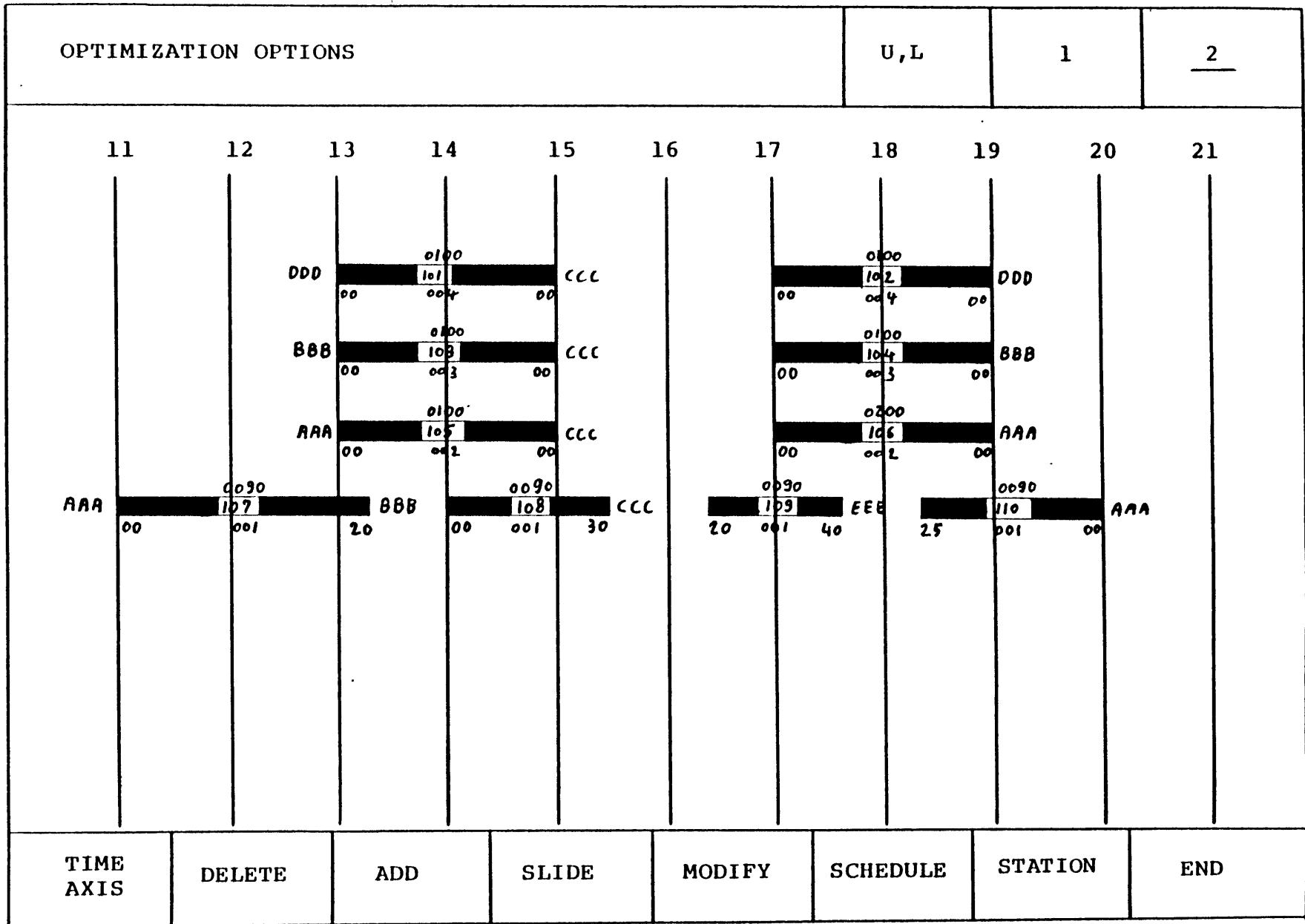


Figure 49 - Flights for aircraft in a fleet of four

OPTIMIZATION OPTIONS								U,L	1	<u>2</u>
11	12	13	14	15	16	17	18	19	20	21
		AAA	0200 101	BBB		0200 102	AAA			
		CCC	0200 103	DDD		0200 104	CCC			
		EEE	0200 105	FFF		0200 106	EEE			
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END			

06

Figure 50 - Flight of one aircraft. AAA-BBB-AAA and CCC-DDD-CCC also qualify

OPTIMIZATION OPTIONS						U,L	1	2
<p>imbalanced stations: EEE, CCC</p> <p>mouse to continue</p>								
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END	

Figure 51 - Attempt to delete flight segment 109 leads
to an imbalanced flow in the network

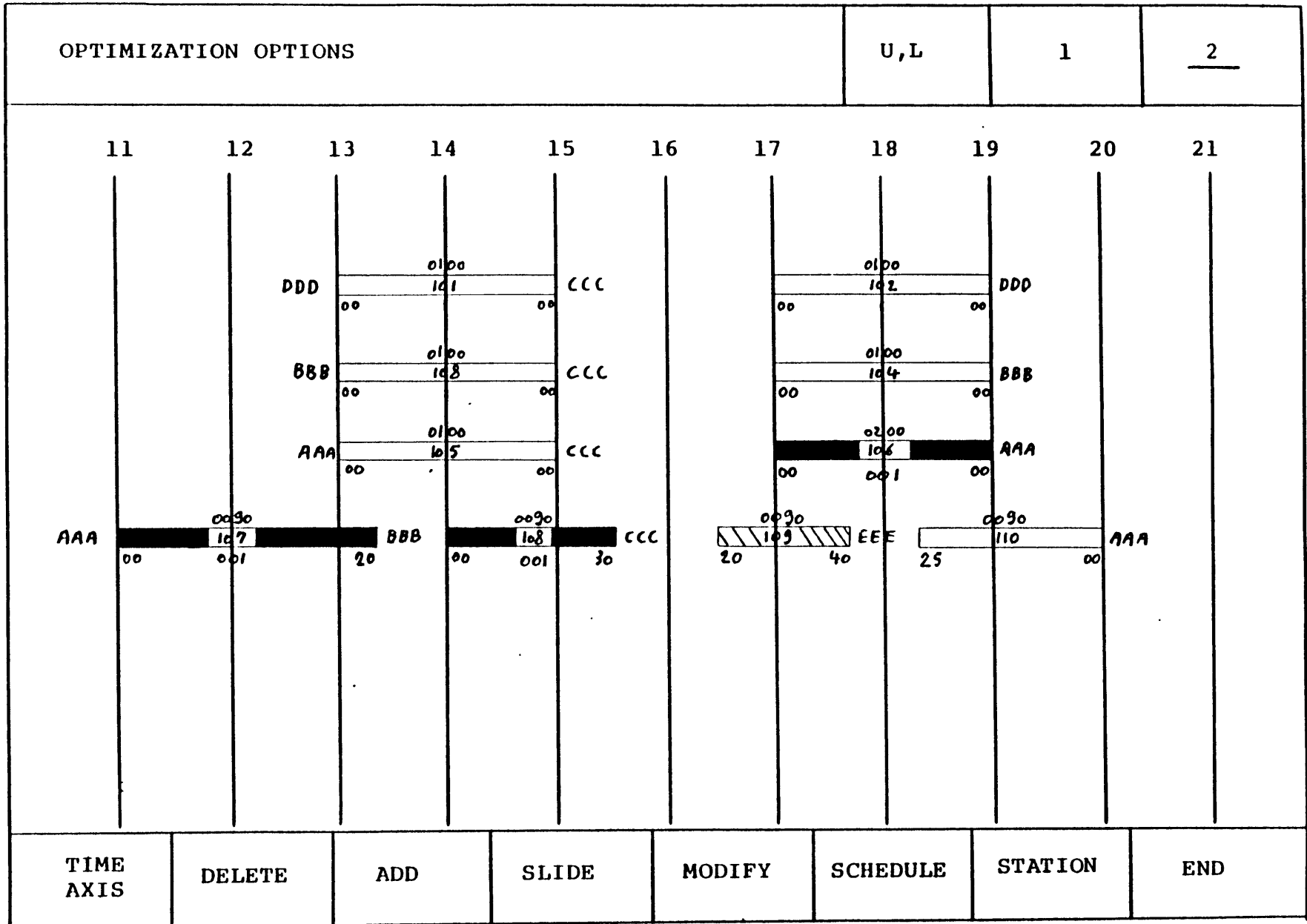


Figure 52 - After flight segment 109 has been "removed", aircraft 001 is rerouted

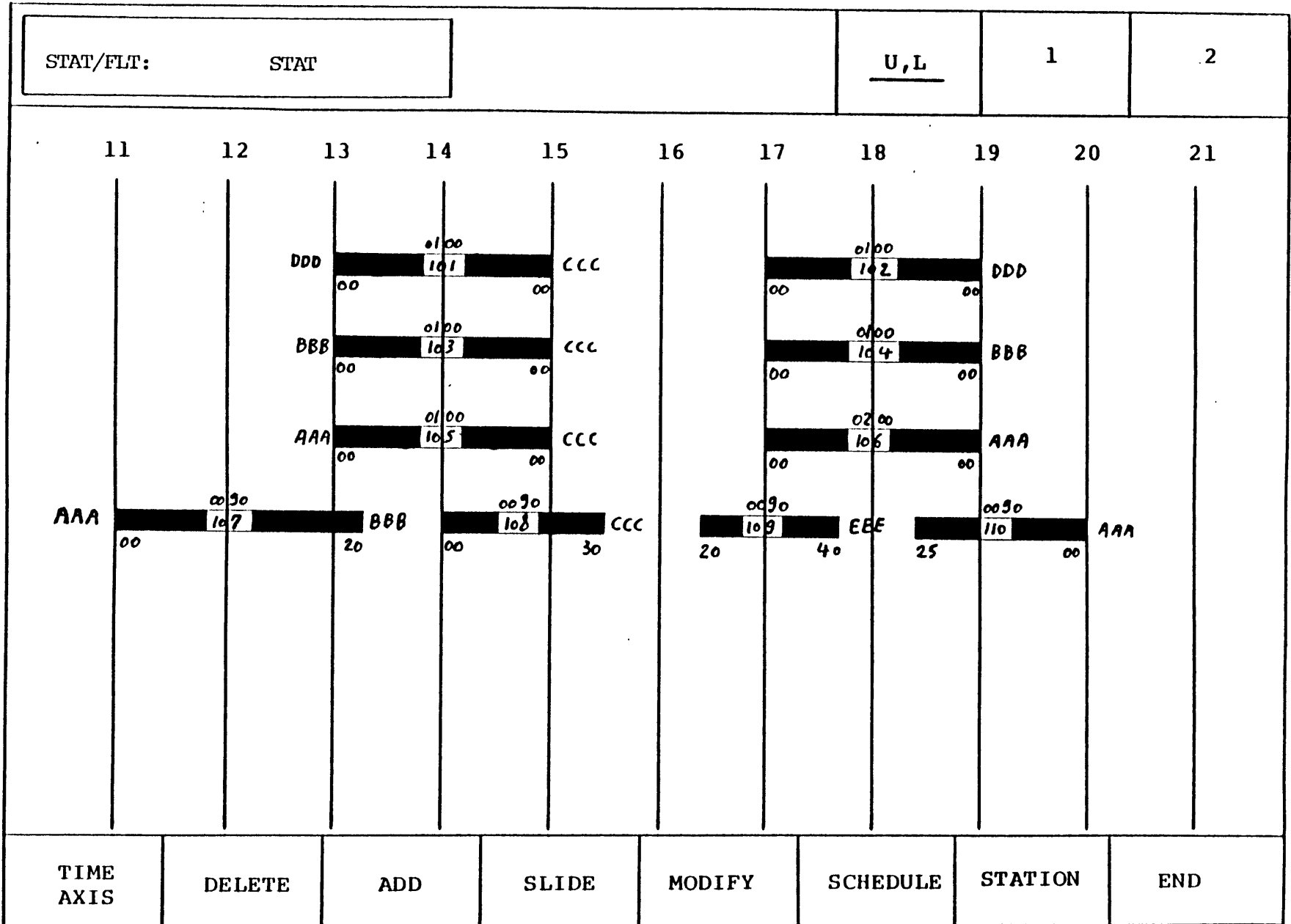
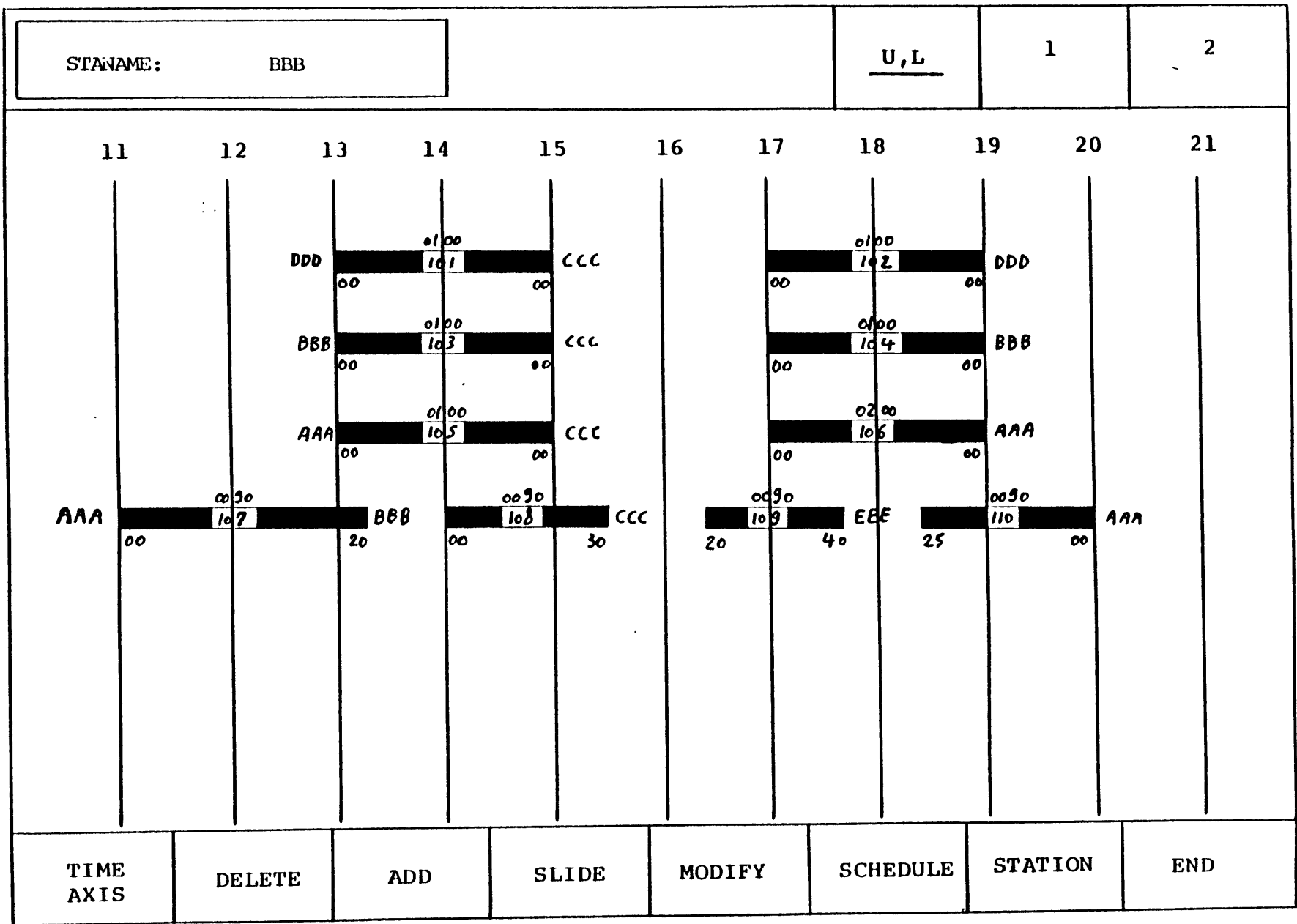


Figure 53 - The flow bounds on a cycle arc are going to be changed



94

Figure 54 - BBB's cycle arc is selected

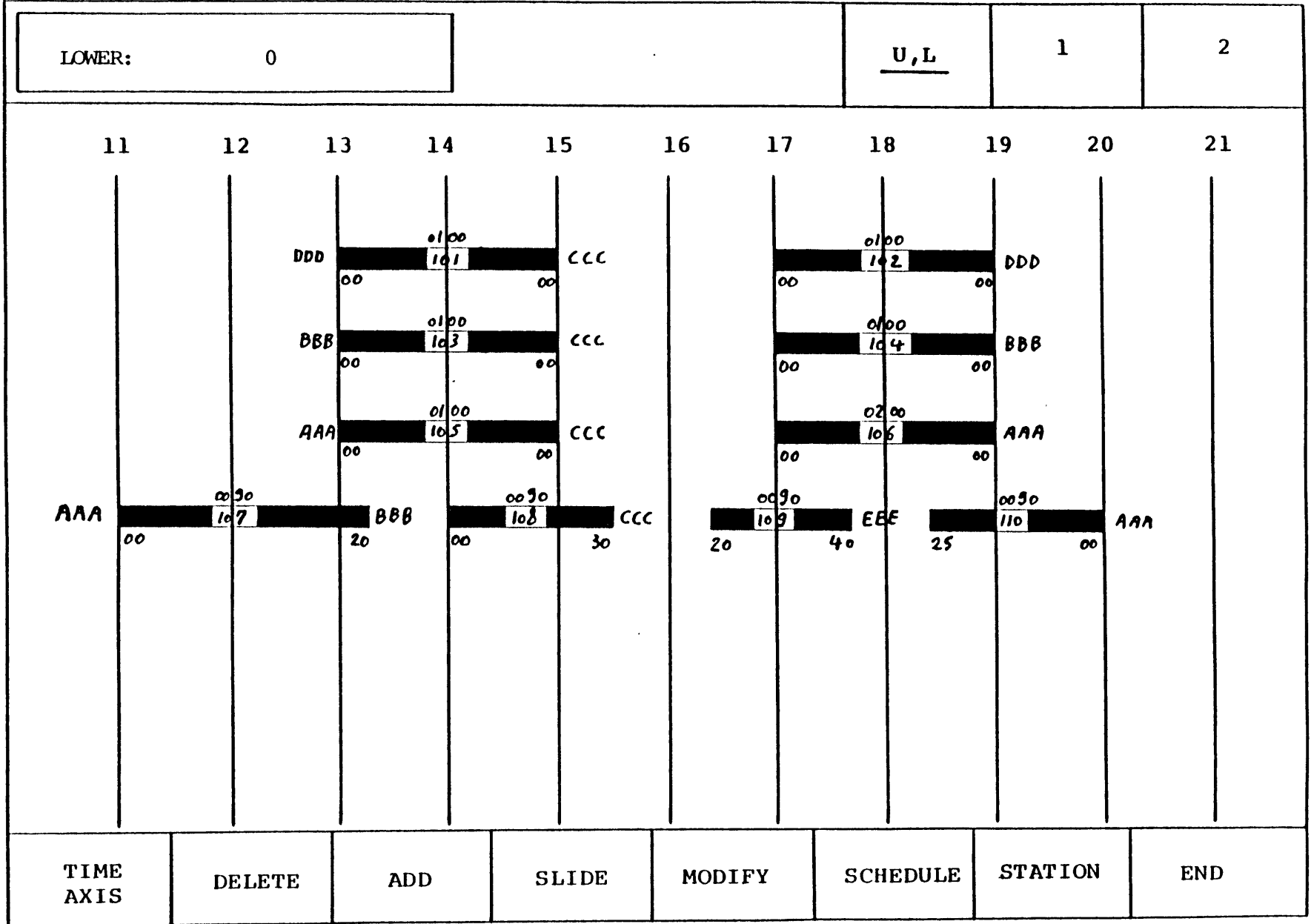


Figure 55 - BBB's cycle arc lower bound remains 0

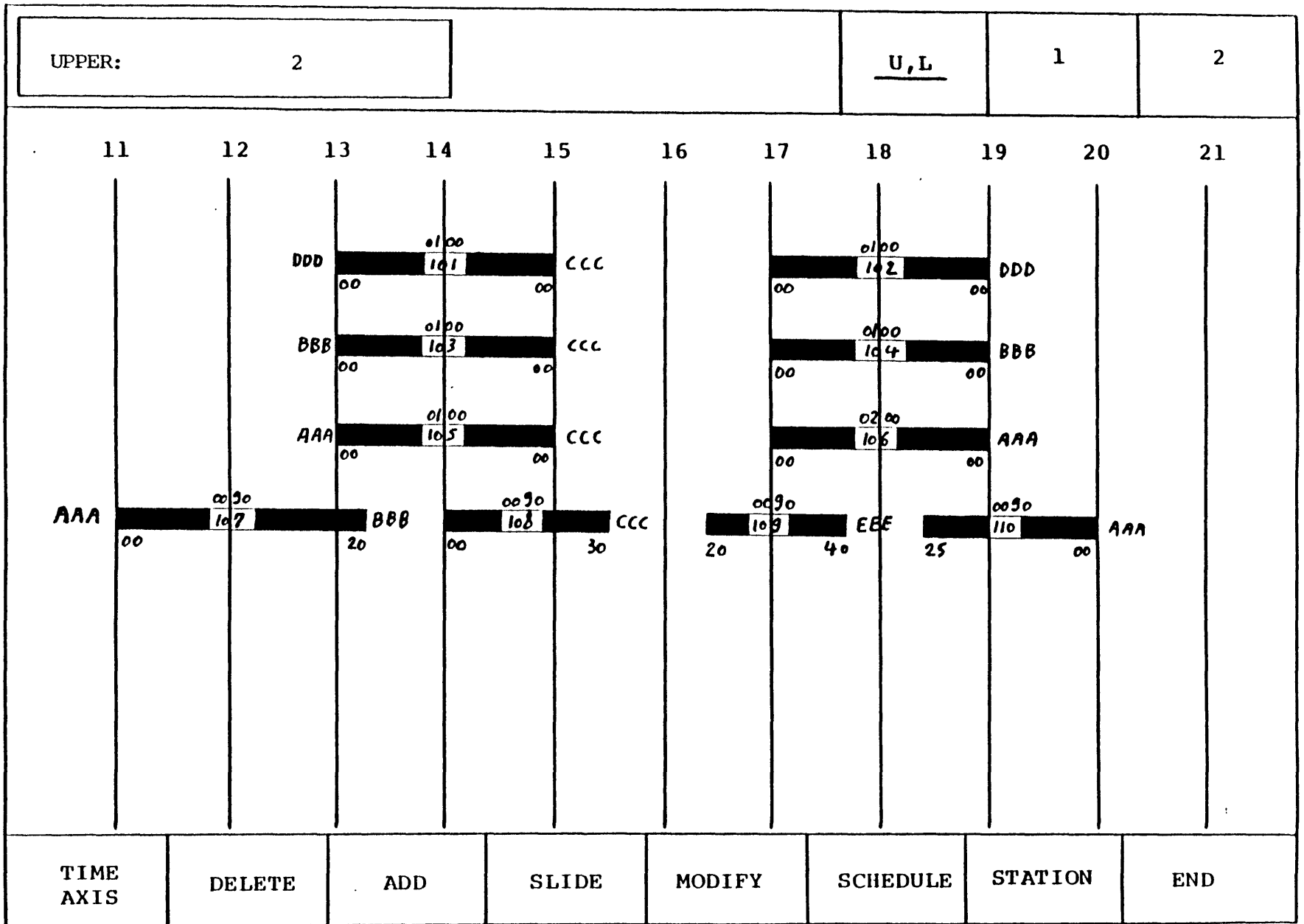


Figure 56 - BBB's cycle arc upper bound becomes 2

97

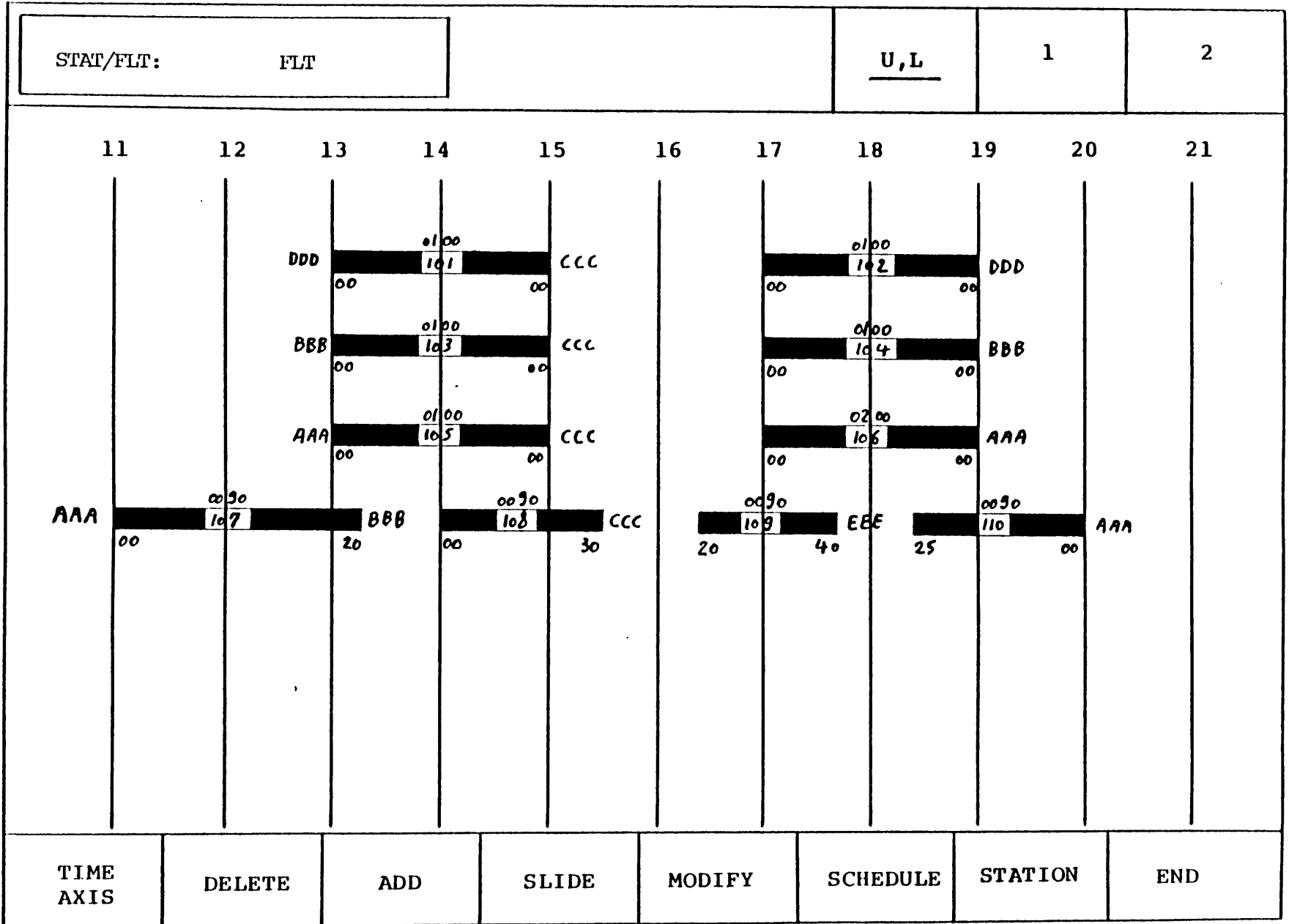
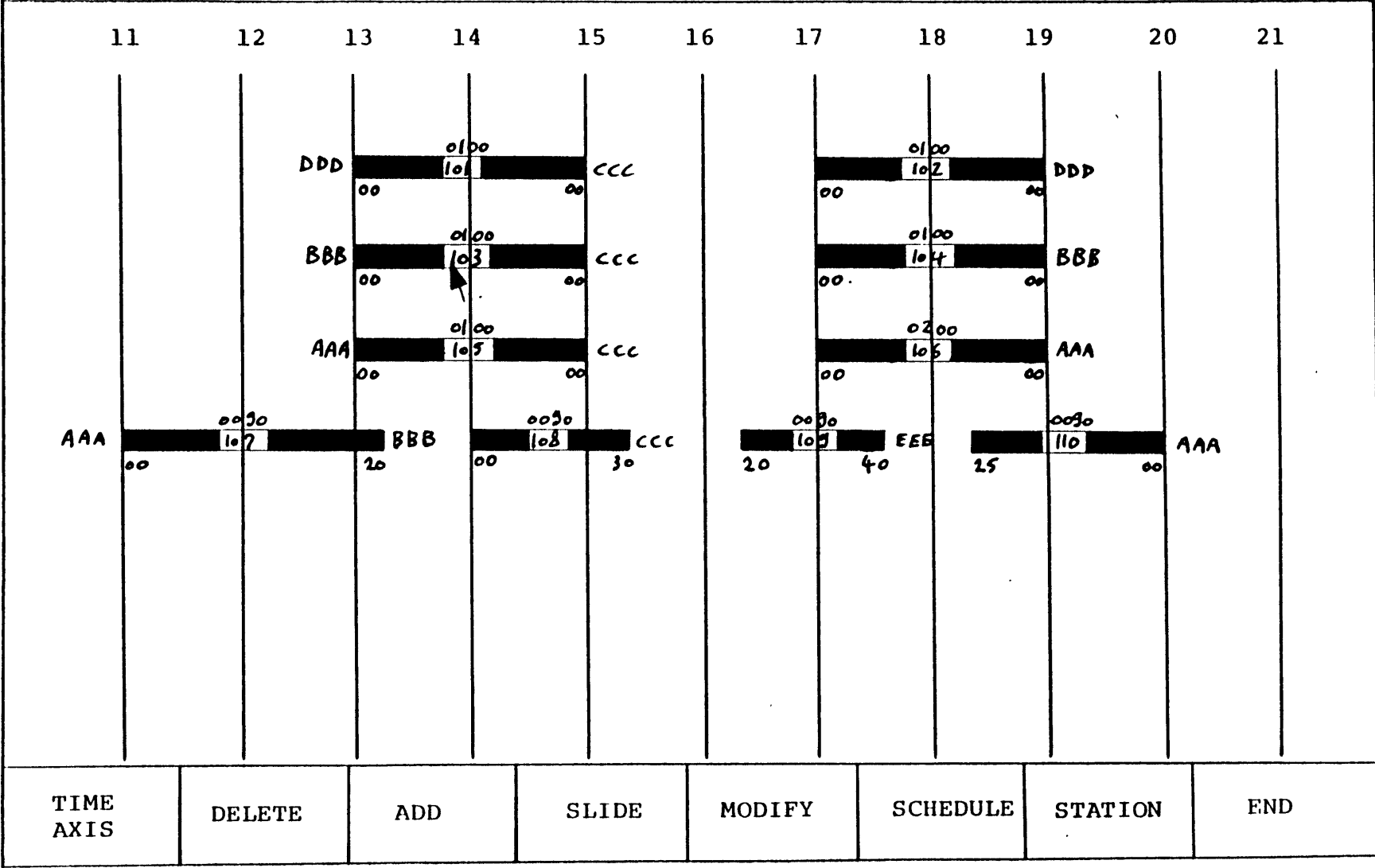


Figure 57 - The flow bounds on a flight arc are going to be changed

SELECT SEGMENT TO BE MODIFIED . . . MOUSE DURING FLYING TIME

51



86

Figure 58 - Flight segment 103 is selected

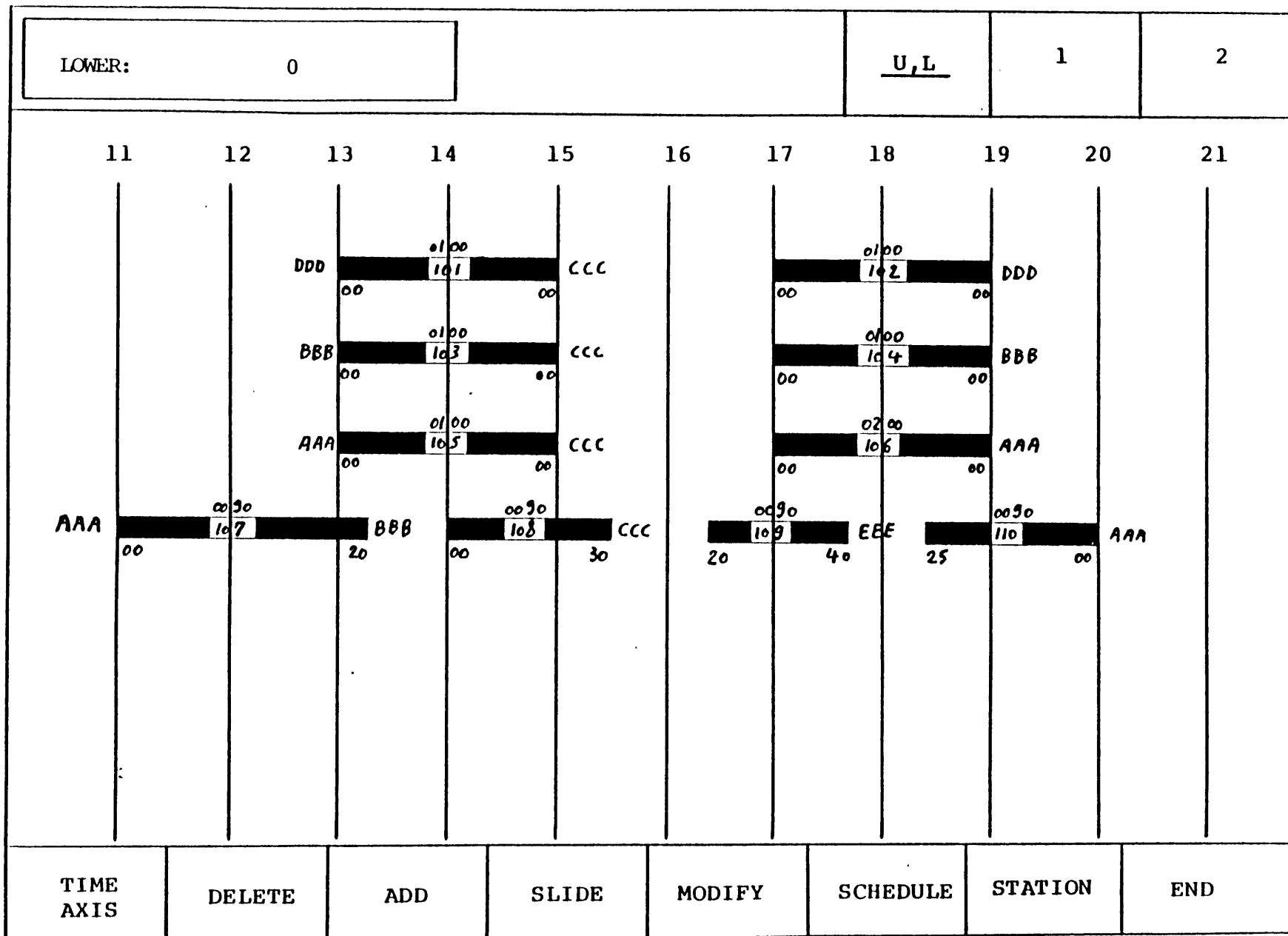
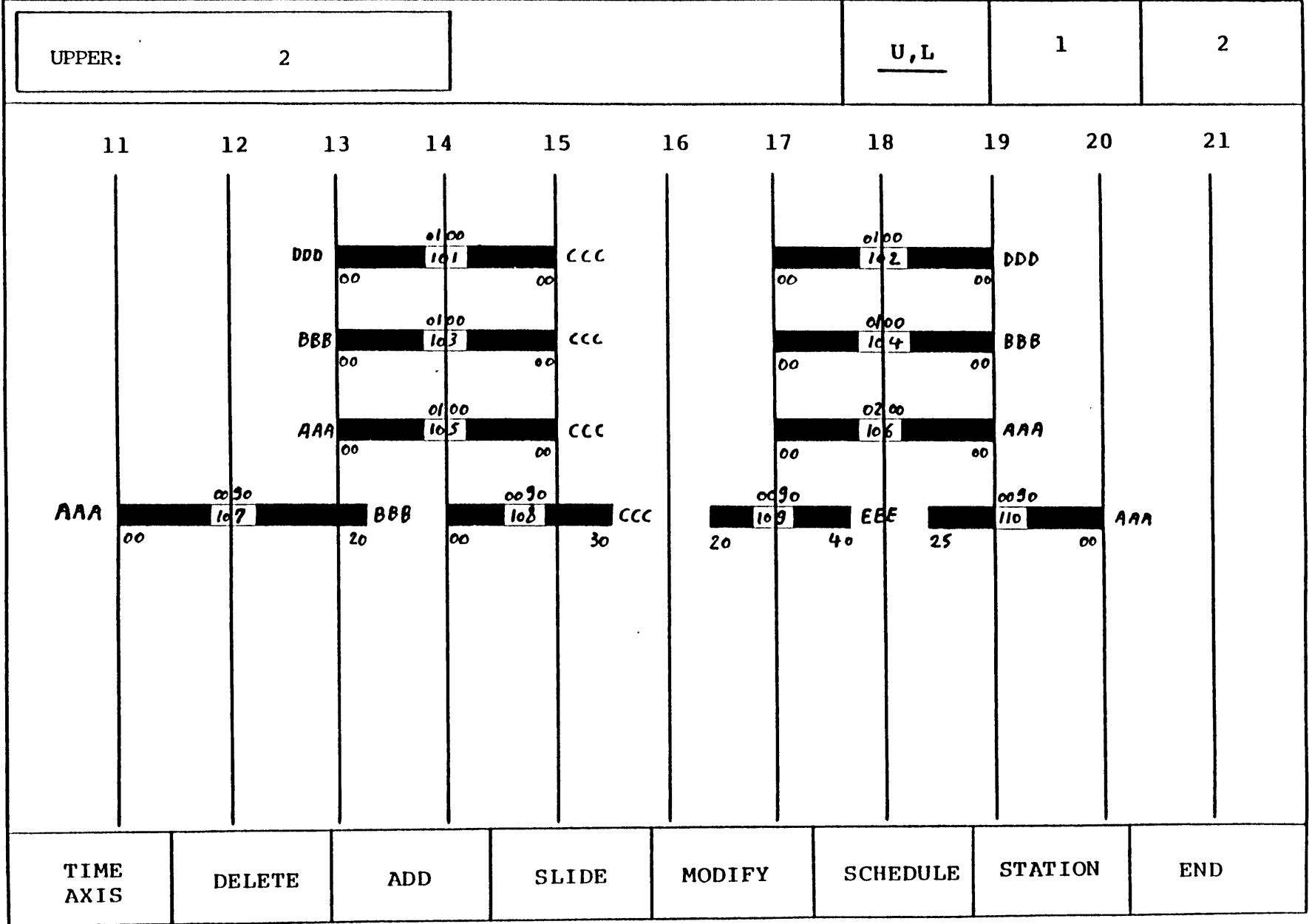


Figure 59 - Flight segment 103's flight arc lower bound
remains at 0



100

Figure 60 - Flight segment 103's flight arc upper bound becomes 2

OPTIMIZATION OPTIONS				U,L	1	<u>2</u>
fleet size: 005 fleet net take: 1760 max doc: -0252 max fleet size: 005 overnight stations: DDD(01), BBB(02), AAA(02)						
grey flights are flown				mouse to continue		
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION END

Figure 61 - Two aircraft now overnight at BBB

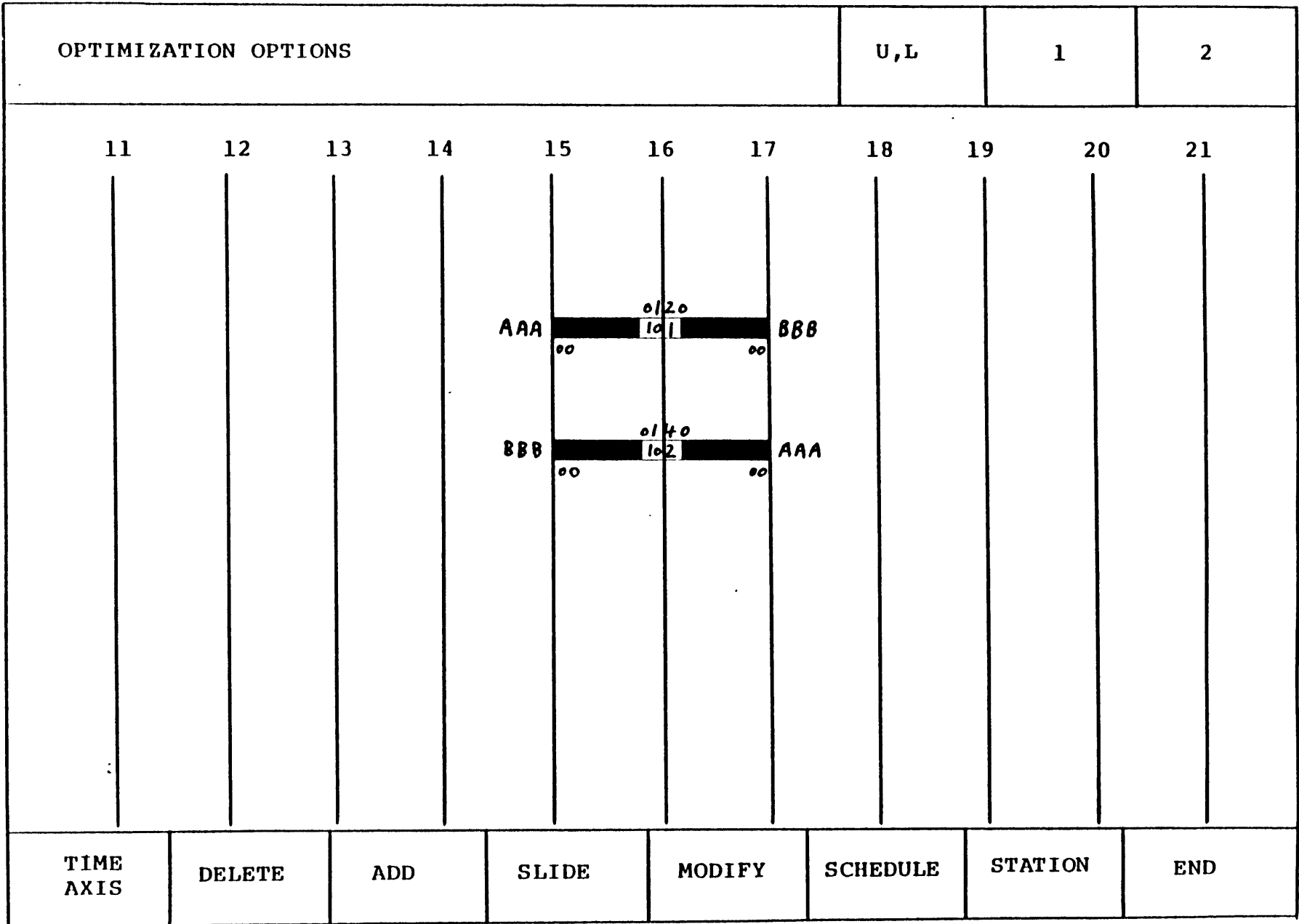


Figure 62 - A "shuttle" schedule plan between AAA and BBB

OPTIMIZATION OPTIONS				U,L	1	<u>2</u>
fleet size: 002 fleet net take: 0060 max doc: -0130 max fleet size: 002 overnight stations: BBB(01), AAA(01)						
grey flights are flown				mouse to continue		
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION END

Figure 63 - Two aircraft are needed to serve the shuttle

OPTIMIZATION OPTIONS								U,L	1	<u>2</u>
11	12	13	14	15	16	17	18	19	20	21
				AAA	0120 101 001	BBB				
				BBB	0140 102 001	AAA				
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 64 - Only one aircraft seems to be required

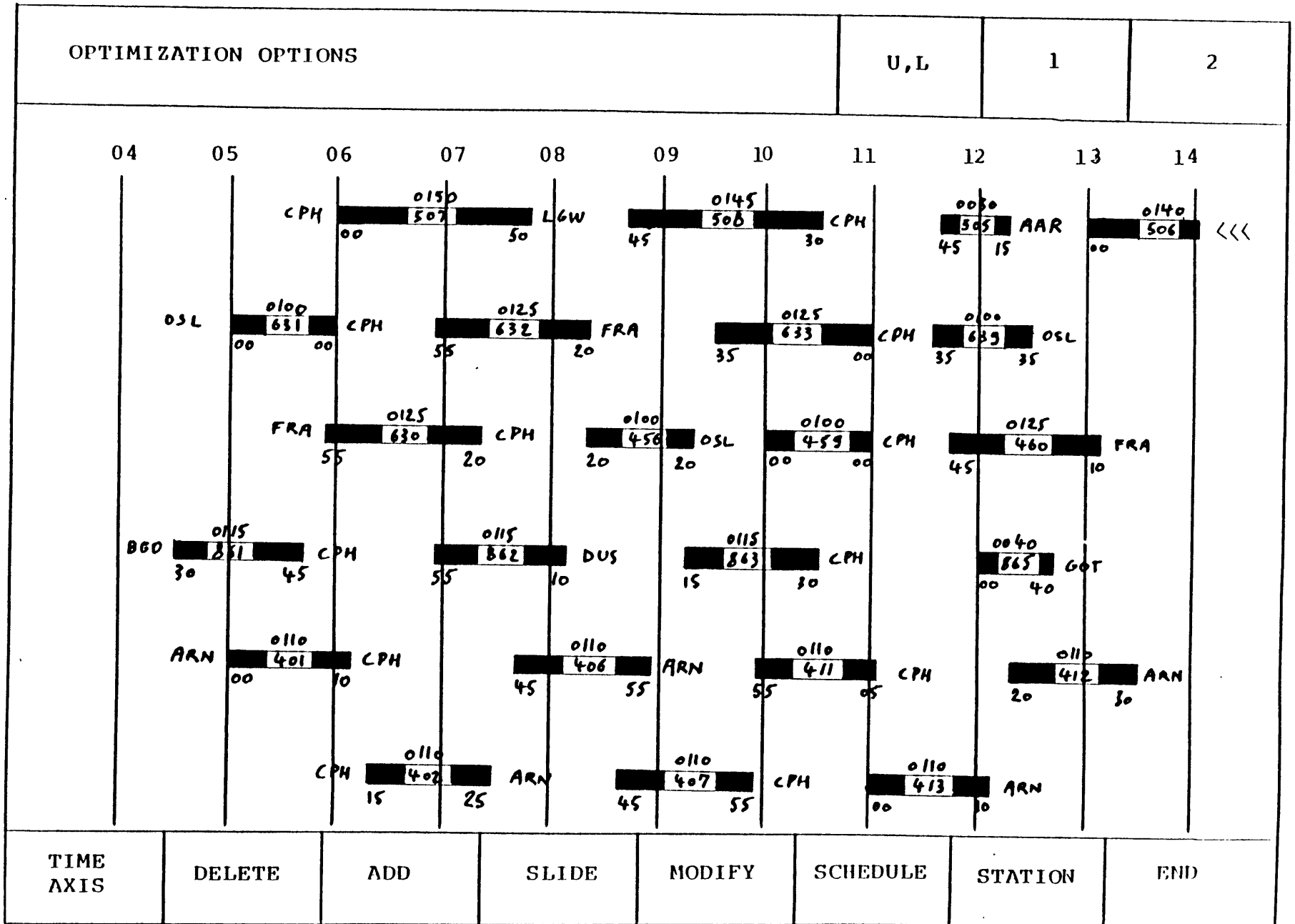


Figure 65 - First part of a SAS schedule plan

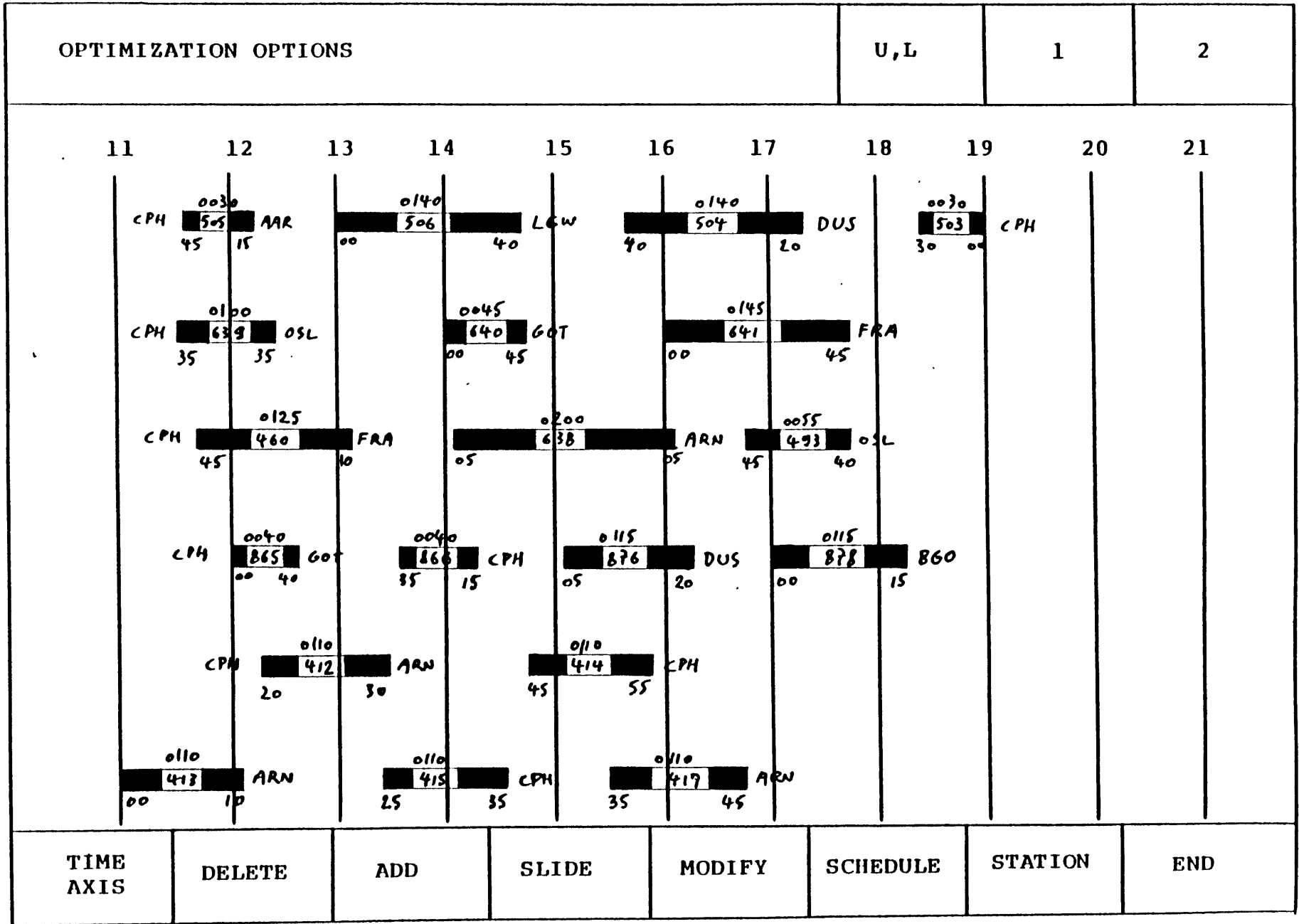


Figure 66 - Second part of a SAS schedule plan

OPTIMIZATION OPTIONS				U,L	<u>1</u>	2	
fleet size: 006 overnight stations: OSL(01), FRA(01), CPH(02), BGO(01), ARN(01)							
mouse to continue							
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END

Figure 67 - Six aircraft are needed to serve all flight segments

OPTIMIZATION OPTIONS				U,L	1	<u>2</u>
fleet size: 006 fleet net take: 4335 max doc: -622 max fleet size: 006 overnight stations: OSL(01), FRA(01), CPH(02), BGO(01), ARN(01)						
grey flights are flown				mouse to continue		
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION END

Figure 68 - Answer to the second optimization option
with fs=6 and doc=-100

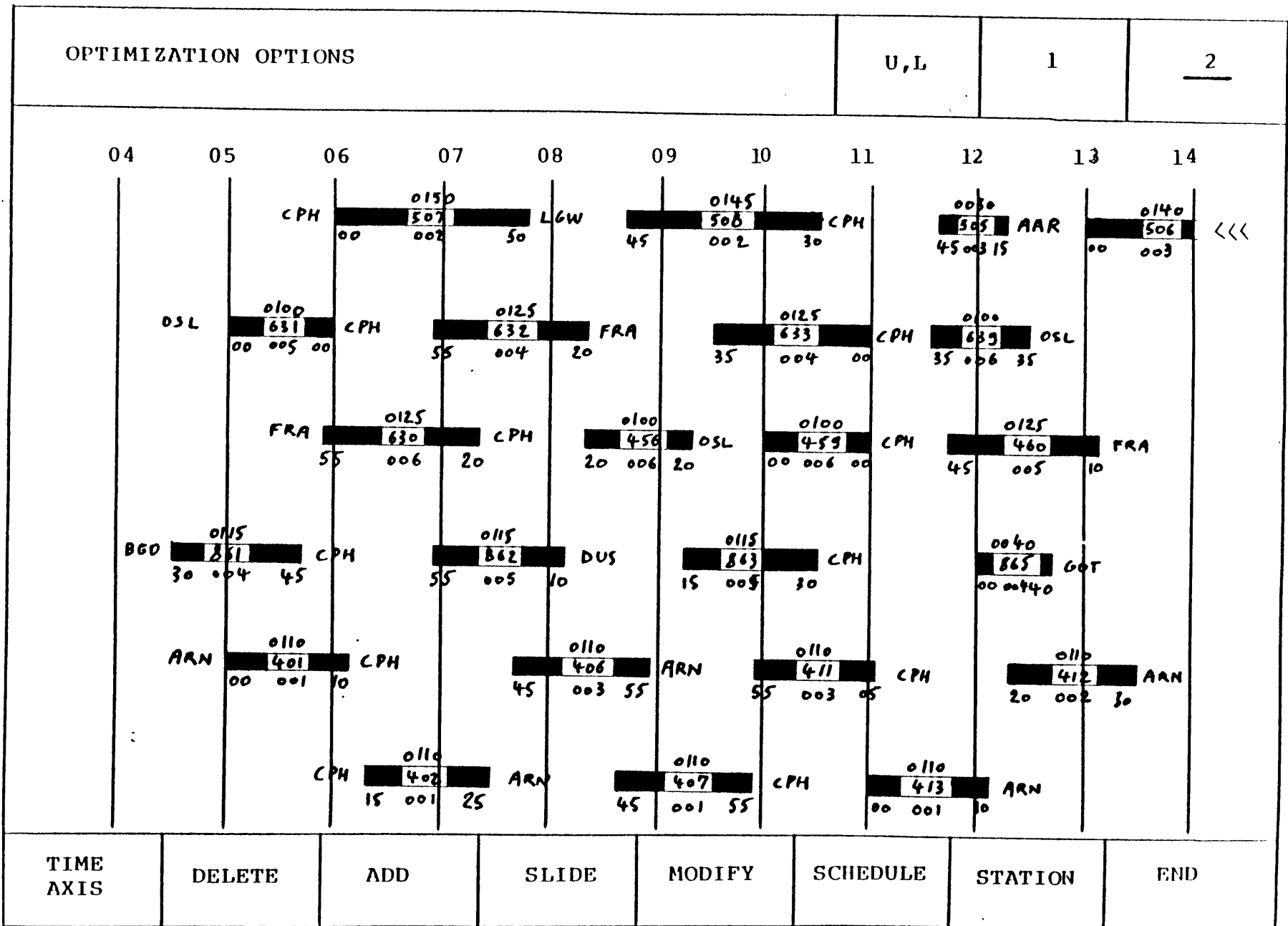


Figure 69 - Aircraft flights with fs=6 (first part)

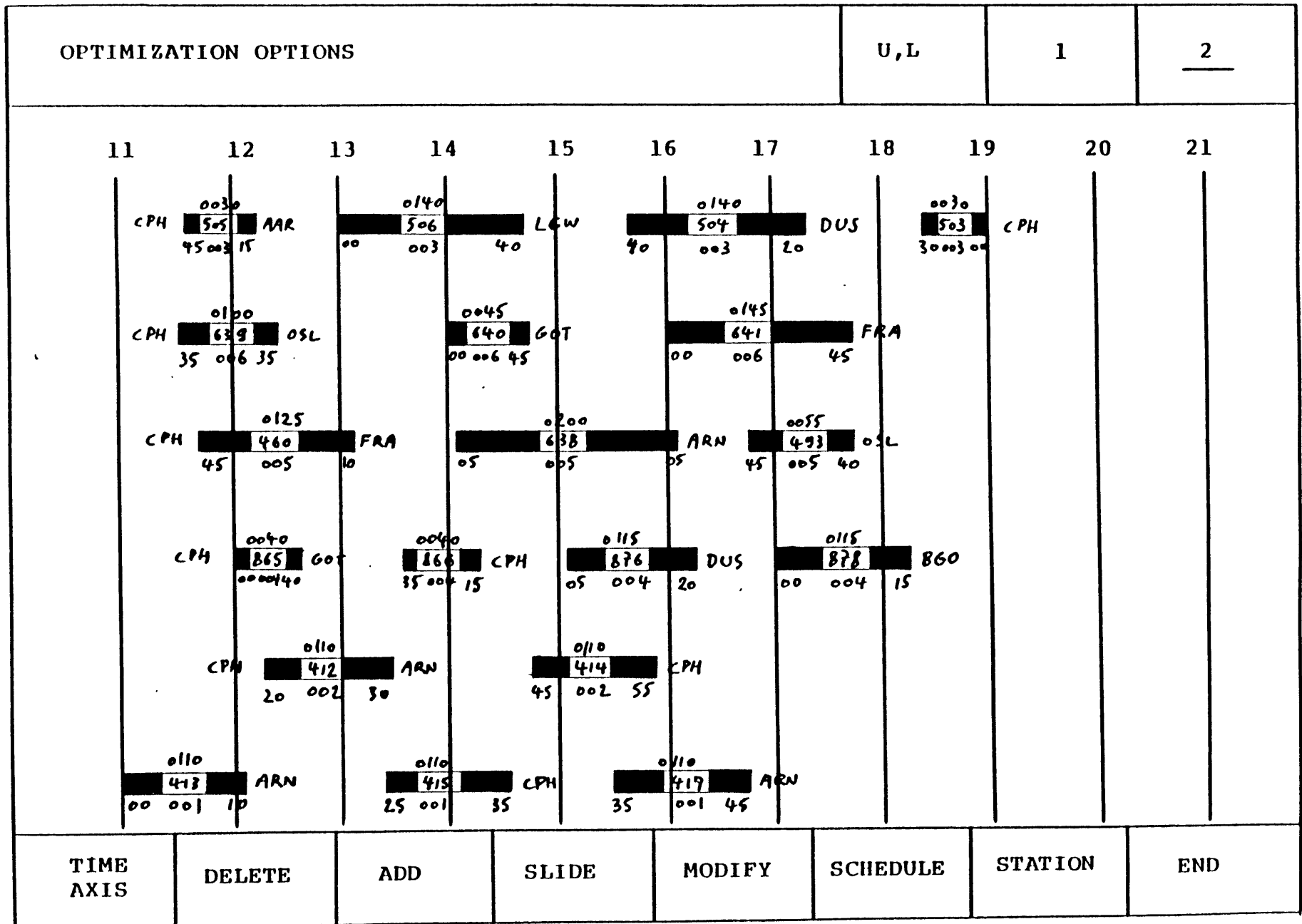


Figure 70 - Aircraft flights with fs=6 (second part)

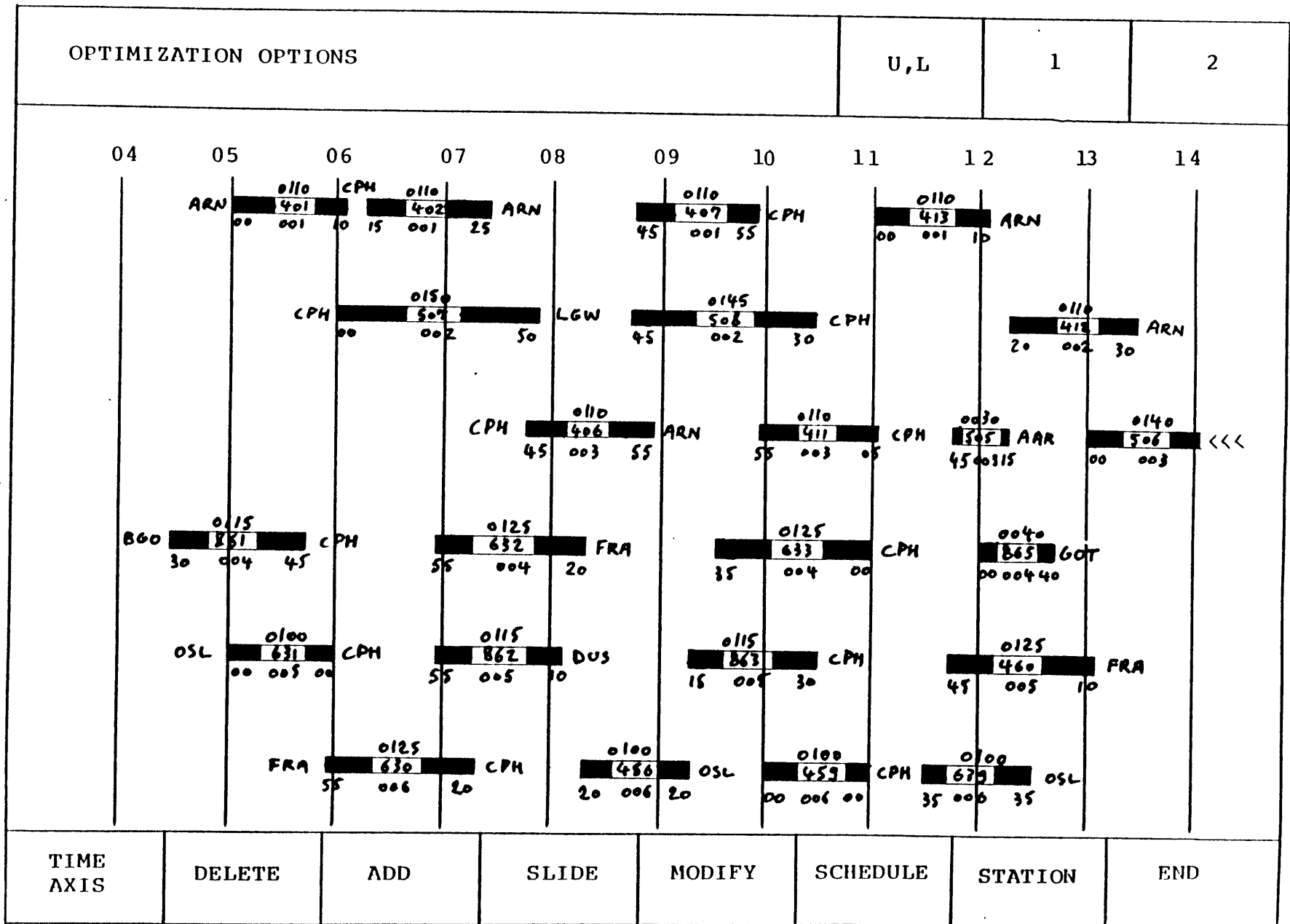


Figure 71 - The aircraft flights have been manually rearranged (first part)

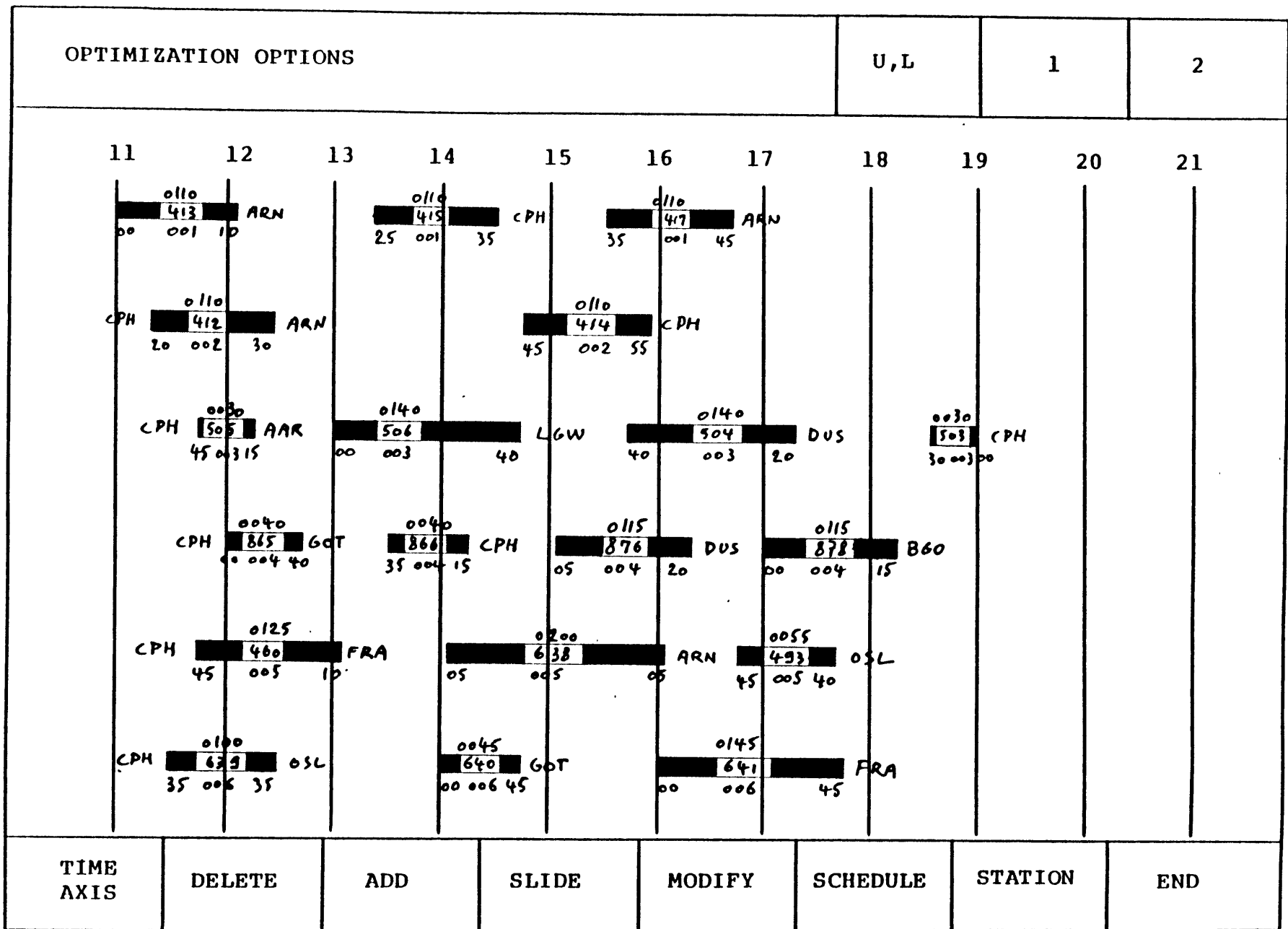


Figure 72 - The aircraft flights have been manually rearranged (second part)

OPTIMIZATION OPTIONS						U,L	1	<u>2</u>
<p>fleet size: 005</p> <p>fleet net take: 3815</p> <p>max doc: - 633</p> <p>max fleet size: 005</p> <p>overnight stations: OSL(01), FRA(01), CPH(01), BGO(01), ARN(01)</p> <p>grey flights are flown mouse to continue</p>								
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END	

Figure 73 - Answer to the second optimization option
with fs=5 and doc=-100

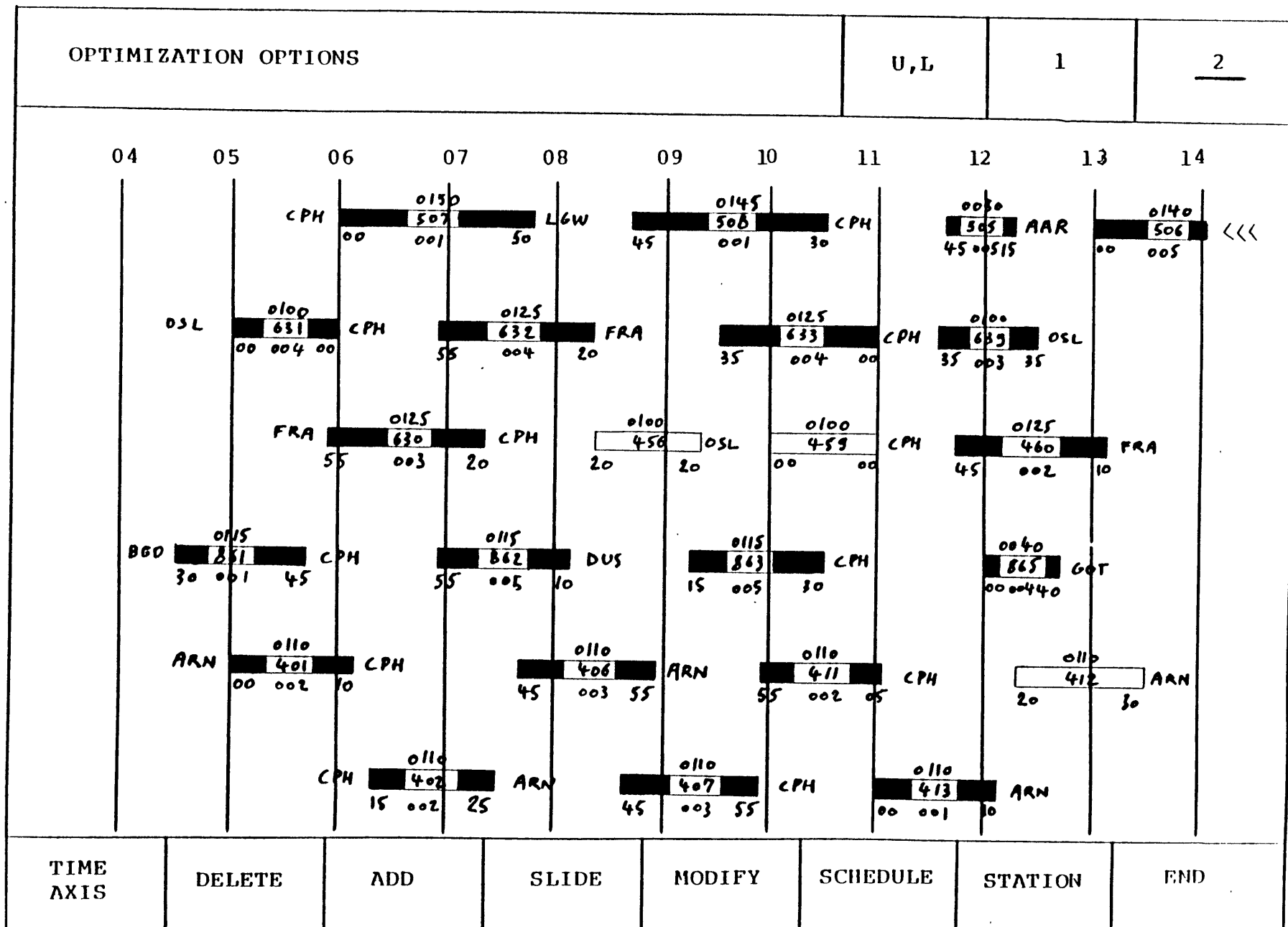


Figure 74 - Aircraft flights with fs=5 (first part)

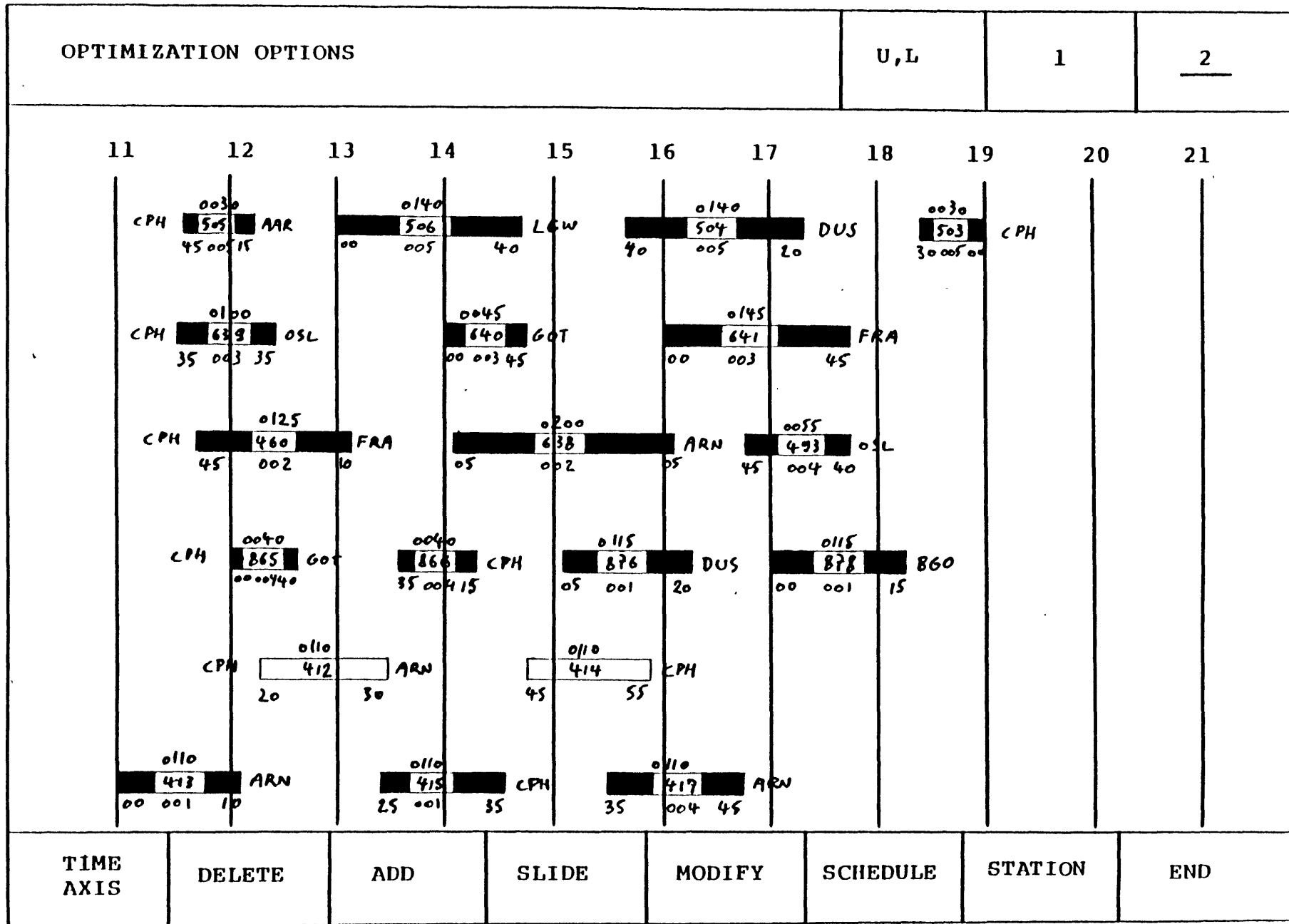


Figure 75 - Aircraft flights with fs=5 (second part)

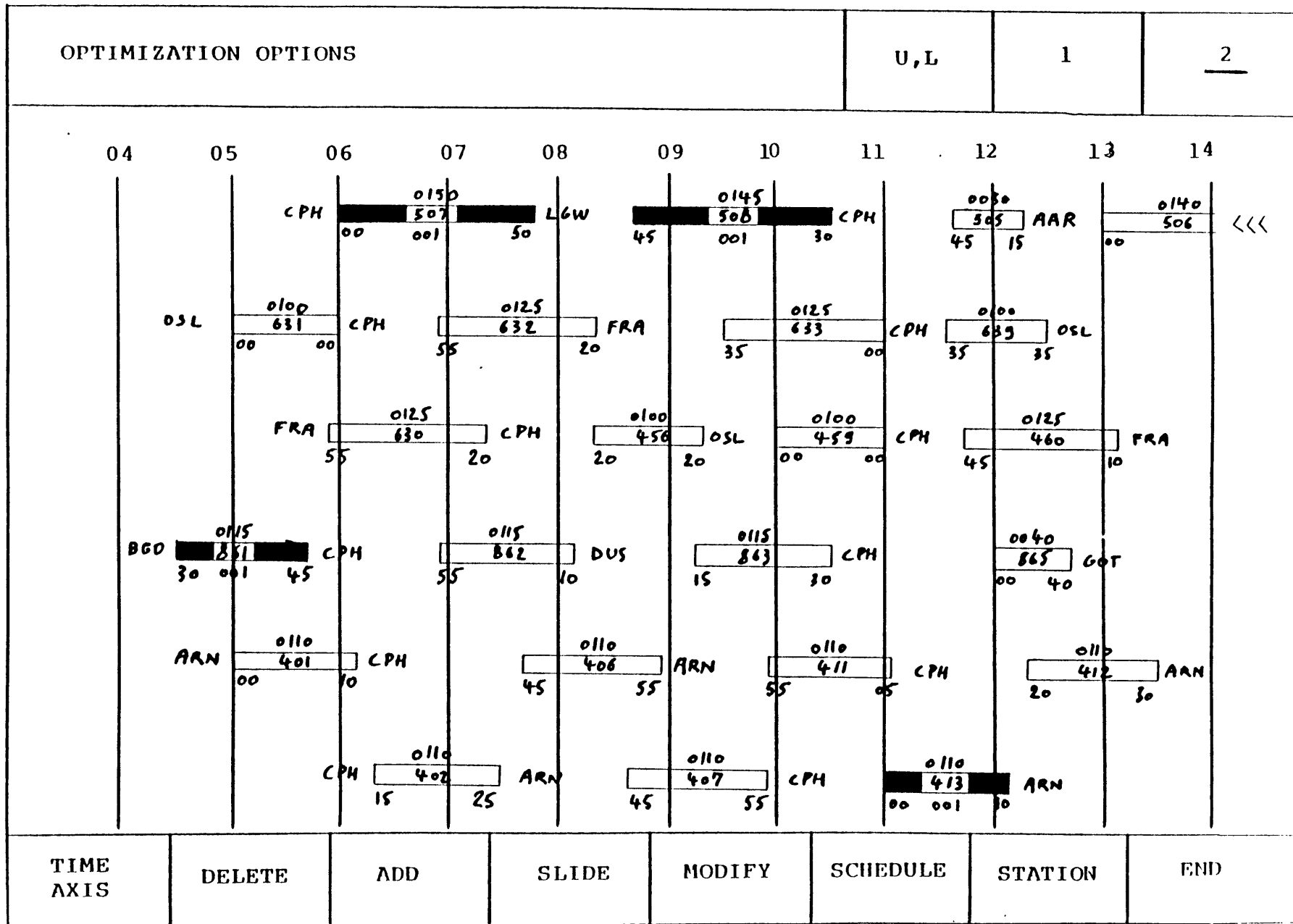


Figure 76 - Aircraft flight with fs=1 (first part)

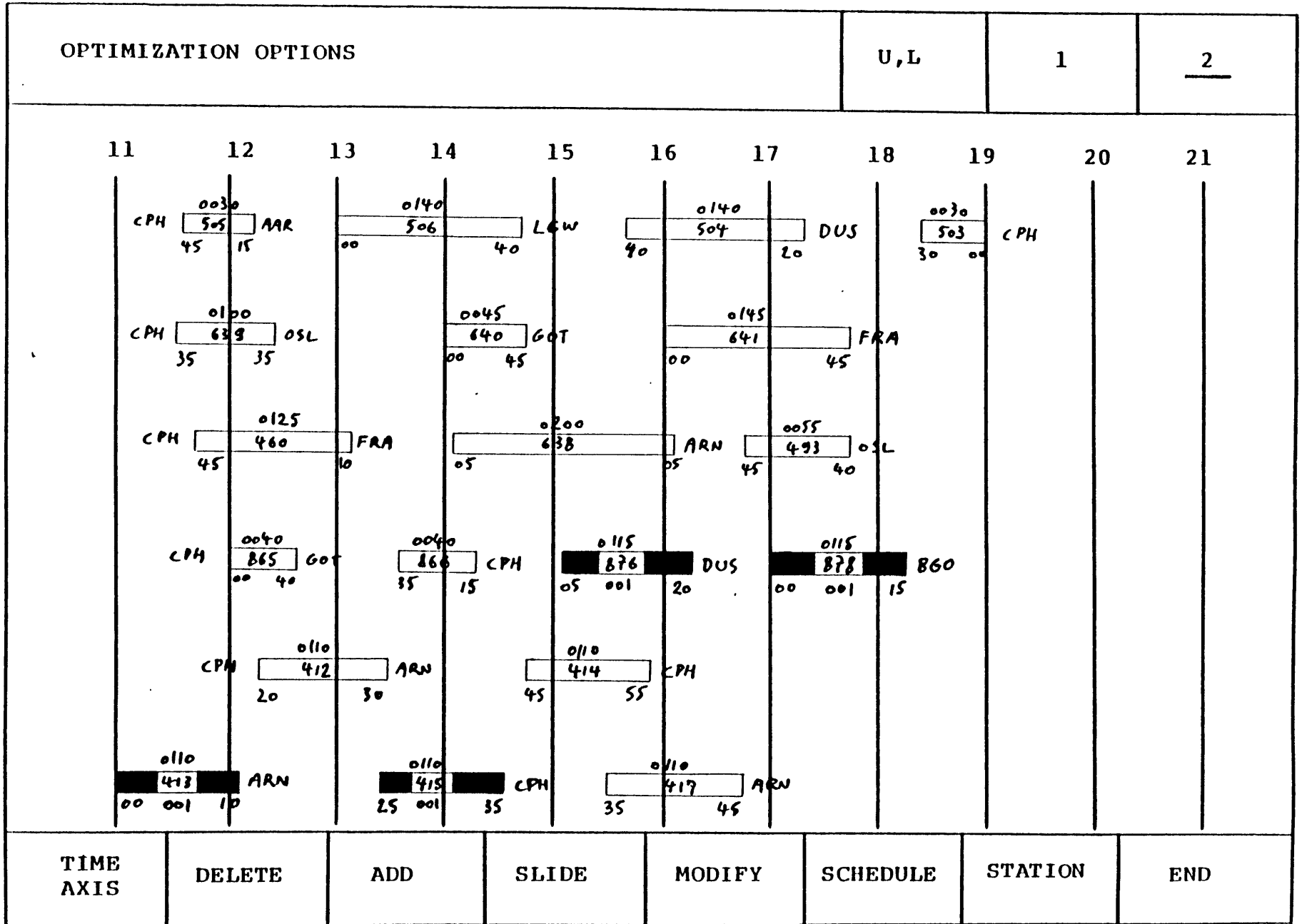


Figure 77 - Aircraft flight with fs=1 (second part)

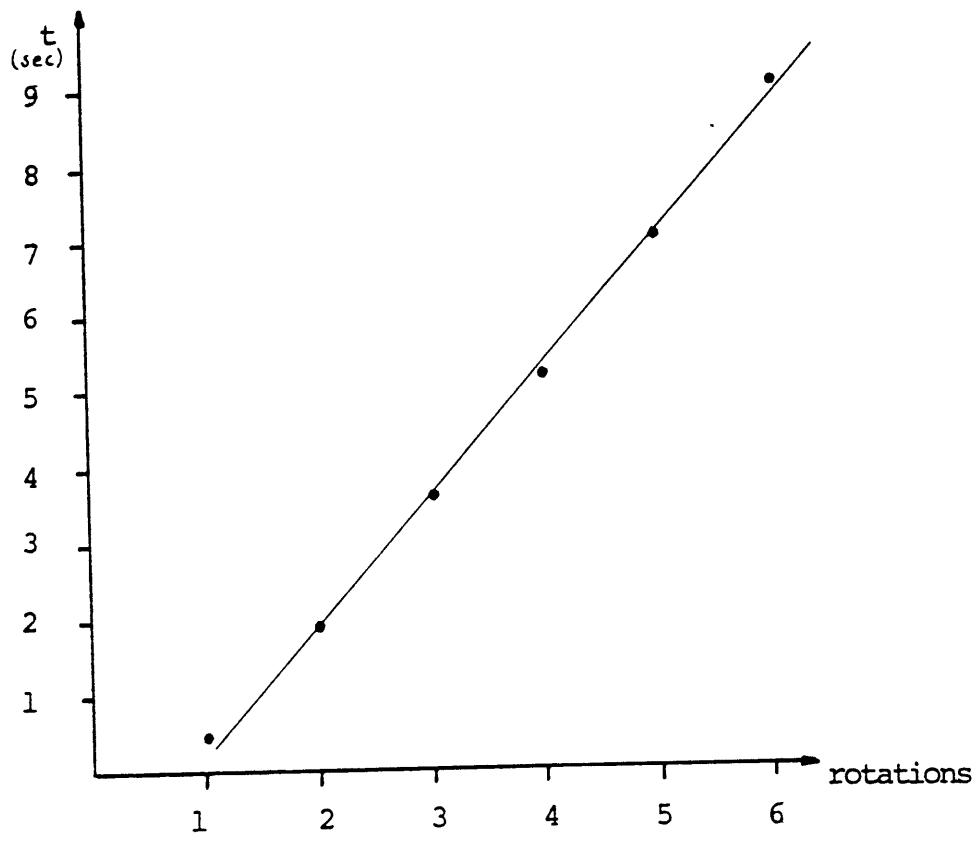


Figure 78 - Change in computation time with number of rotations

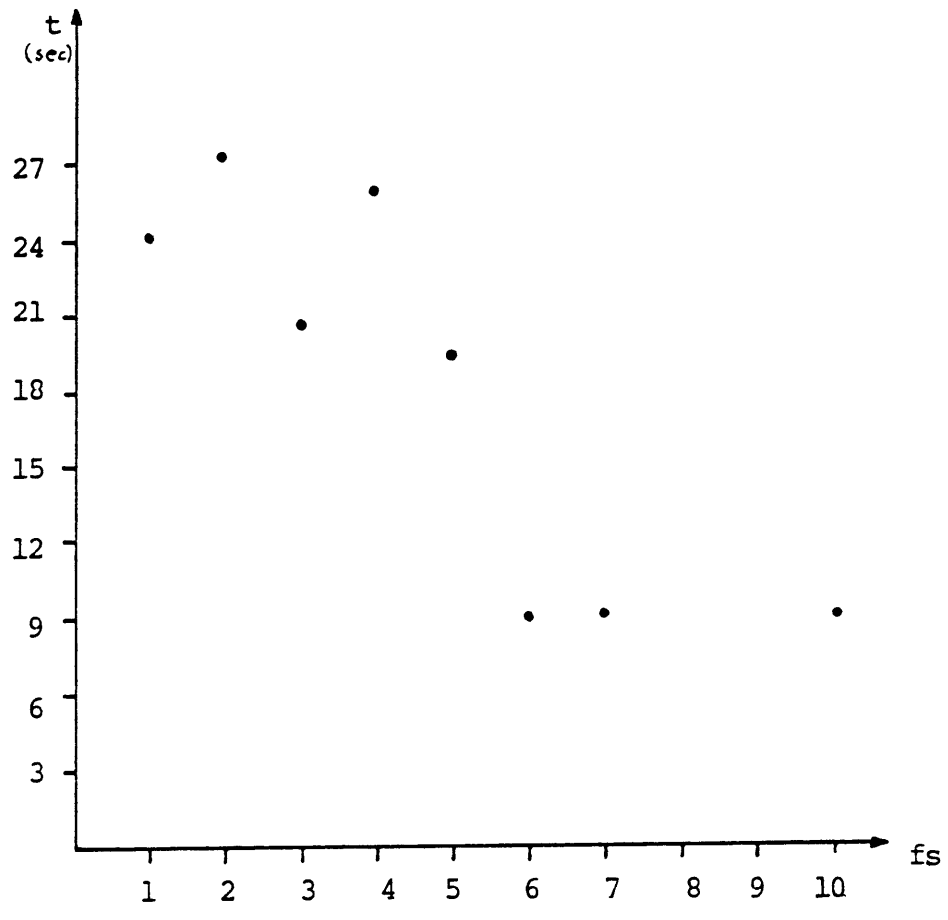


Figure 79 - Change in computation time with fleet size

DAY: 014				DAY				1	2	
11	12	13	14	15	16	17	18	19	20	21
TIME AXIS	DELETE	<u>ADD</u>	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 80 - Input of day of occurrence of a flight segment
of a multiple-day schedule plan

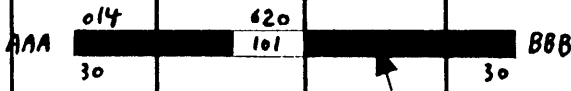
OPTIMIZATION OPTIONS								DAY	1	2
11	12	13	14	15	16	17	18	19	20	21
			AAA	014 30	620 101	BBB	30			
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 81 - Flight segment 101 has been scheduled for day 14

SELECT SEGMENT TO BE MODIFIED . . . MOUSE DURING FLYING TIME

28

11 12 13 14 15 16 17 18 19 20 21



TIME
AXIS

DELETE

ADD

SLIDE

MODIFY

SCHEDULE

STATION

END

Figure 82 - Flight segment 101's day of occurrence will be changed

DAY: 39				<u>DAY</u>				1	2	
11	12	13	14	15	16	17	18	19	20	21
				AAA	014 30	620 101	BBB 30			
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END			

123

Figure 83 - Flight segment 101 is rescheduled for day 39

OPTIMIZATION OPTIONS								DAY	1	2
11	12	13	14	15	16	17	18	19	20	21
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 84 - Flight segment 101 has been scheduled for day 39

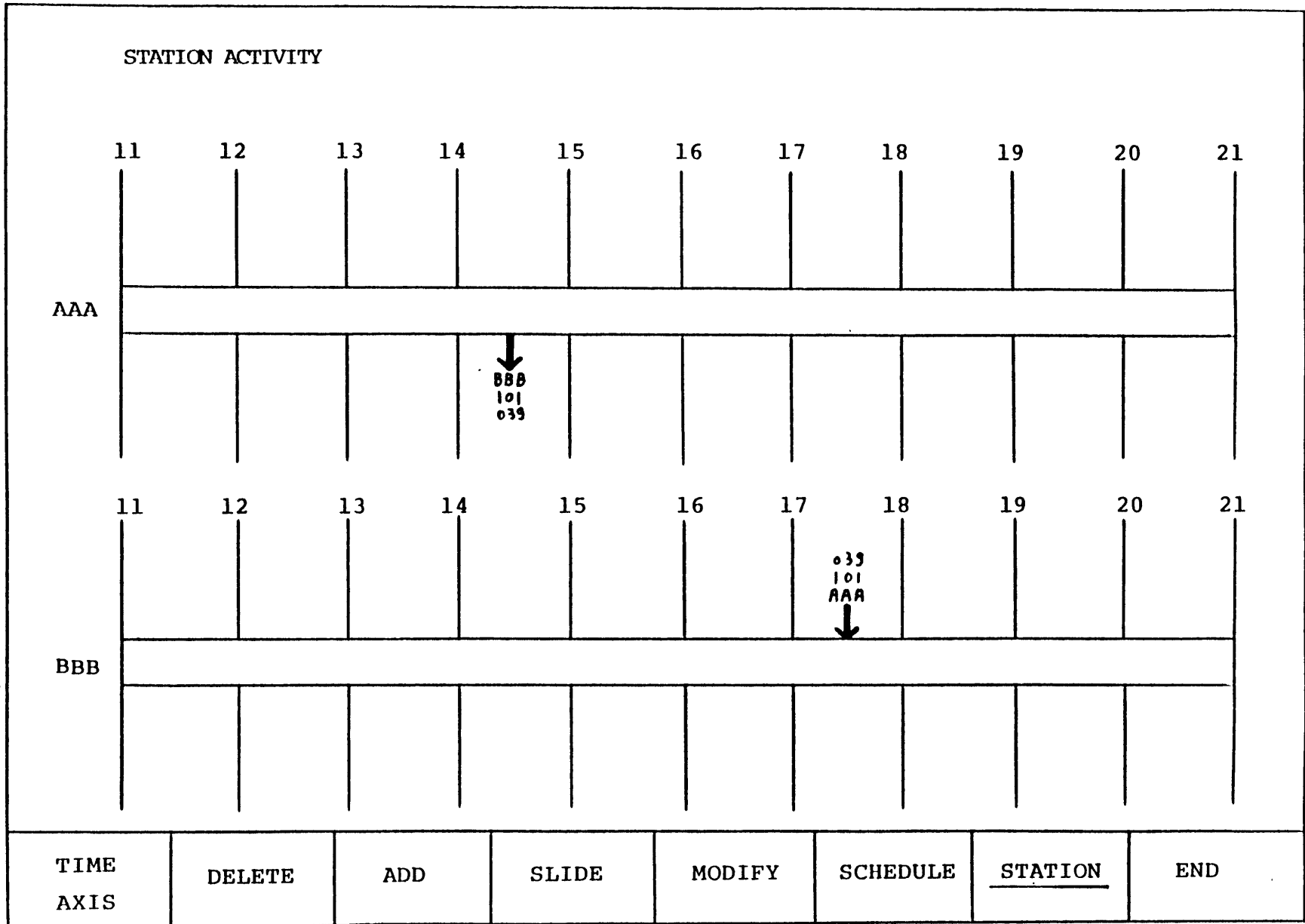


Figure 85 - Station display for flight segment 101

OPTIMIZATION OPTIONS								DAY	1	<u>2</u>
11	12	13	14	15	16	17	18	19	20	21
			AAA	012 00	0100 101 001	BBB				
			BBB	021 00	0200 102 001	DDD				
			DDD	029 00	0100 103 001	FFF				
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 86 - One aircraft flies from AAA to FFF in 17 days

OPTIMIZATION OPTIONS				DAY	<u>1</u>	2	
fleet size: 001				mouse to continue			
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END

Figure 87 - Answer to the first optimization option

OPTIMIZATION OPTIONS				DAY	1	<u>2</u>	
fleet size: 001 fleet net take: 0300 max doc: -400 max fleet size: 001							
grey flights are flown				mouse to continue			
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END

Figure 88 - Answer to the second optimization option

OPTIMIZATION OPTIONS								DAY	1	<u>2</u>
11	12	13	14	15	16	17	18	19	20	21
		AAA	123 00	0200 101	BBB					
					BBB	162 00	0150 103 001		CCC	
		ZZZ	141 00	0200 102 001	BBB					
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END			

Figure 89 - Flight ZZZ-BBB-CCC has been selected,
but AAA-BBB-CCC is also possible

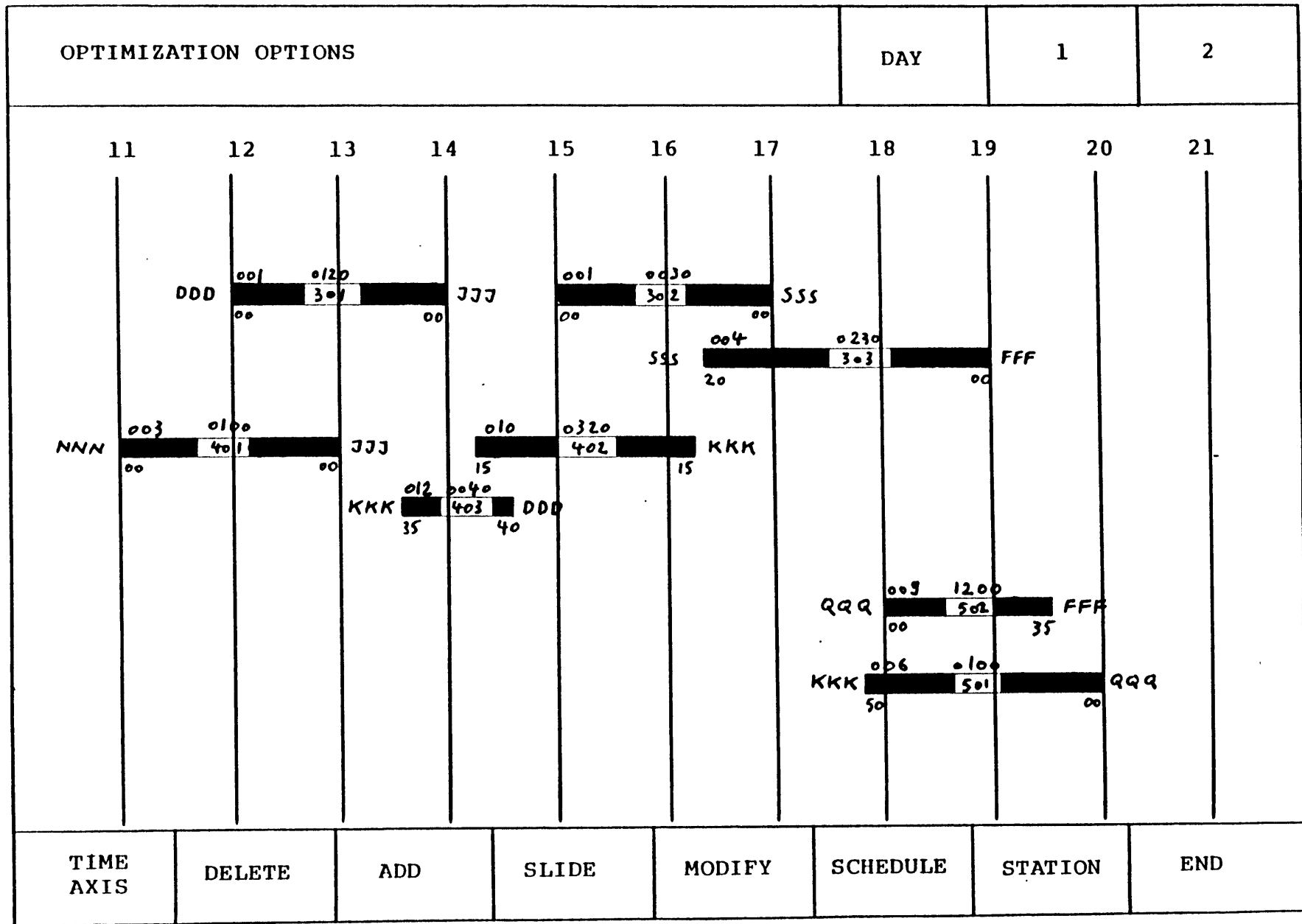


Figure 90 - Multiple-day schedule plan between days 1 and 12

OPTIMIZATION OPTIONS				DAY	1	<u>2</u>	
fleet size: 003 fleet net take: 1840 max doc: - 713 max fleet size: 003 grey flights are flown mouse to continue							
TIME AXIS	DELETE	ADD	SLIDE	MODIFY	SCHEDULE	STATION	END

Figure 91 - Answer to the second optimization option
with fs=3 and doc=-300

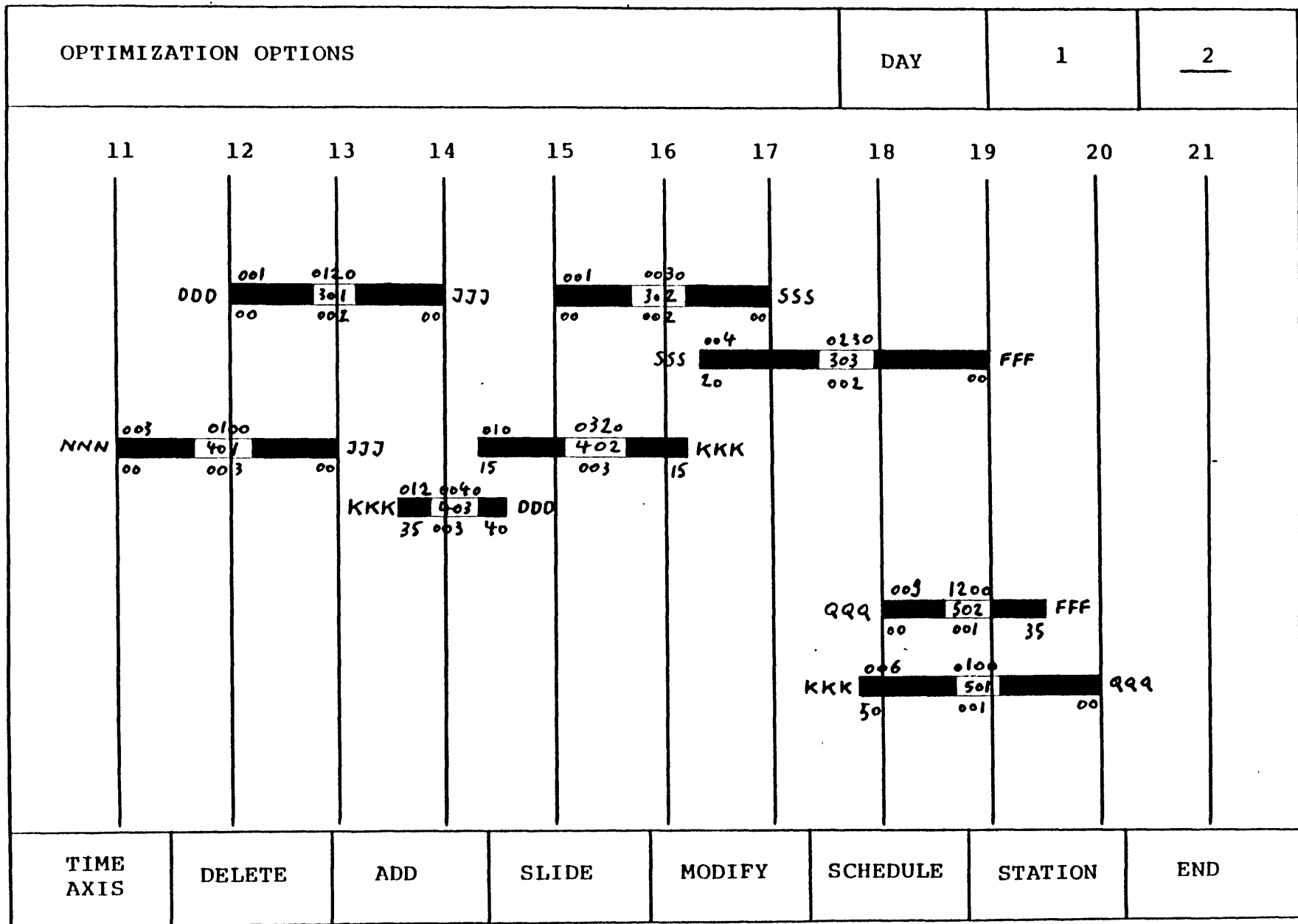


Figure 92 - The flights selected for the three aircraft

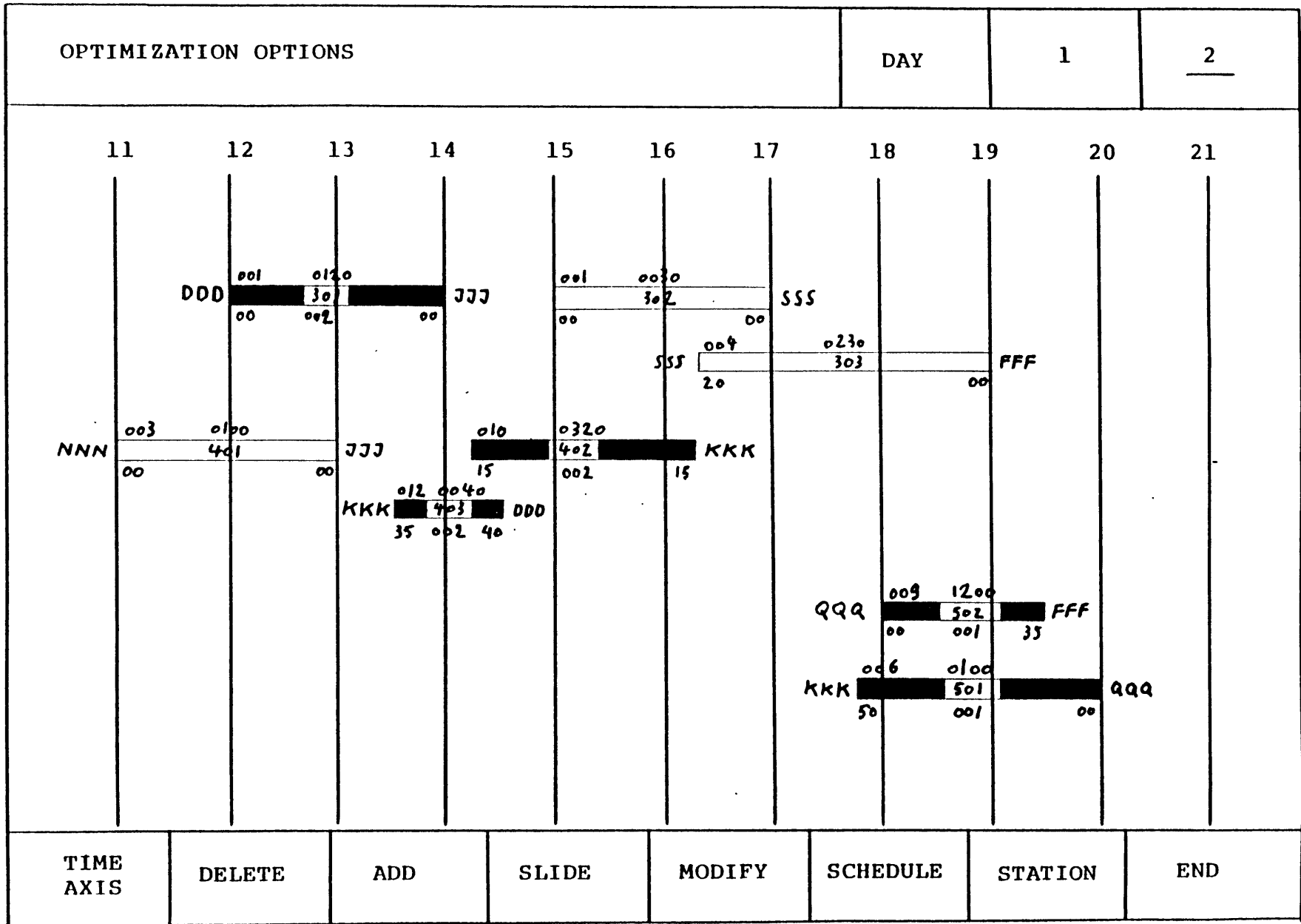


Figure 93 - The flights selected for two aircraft

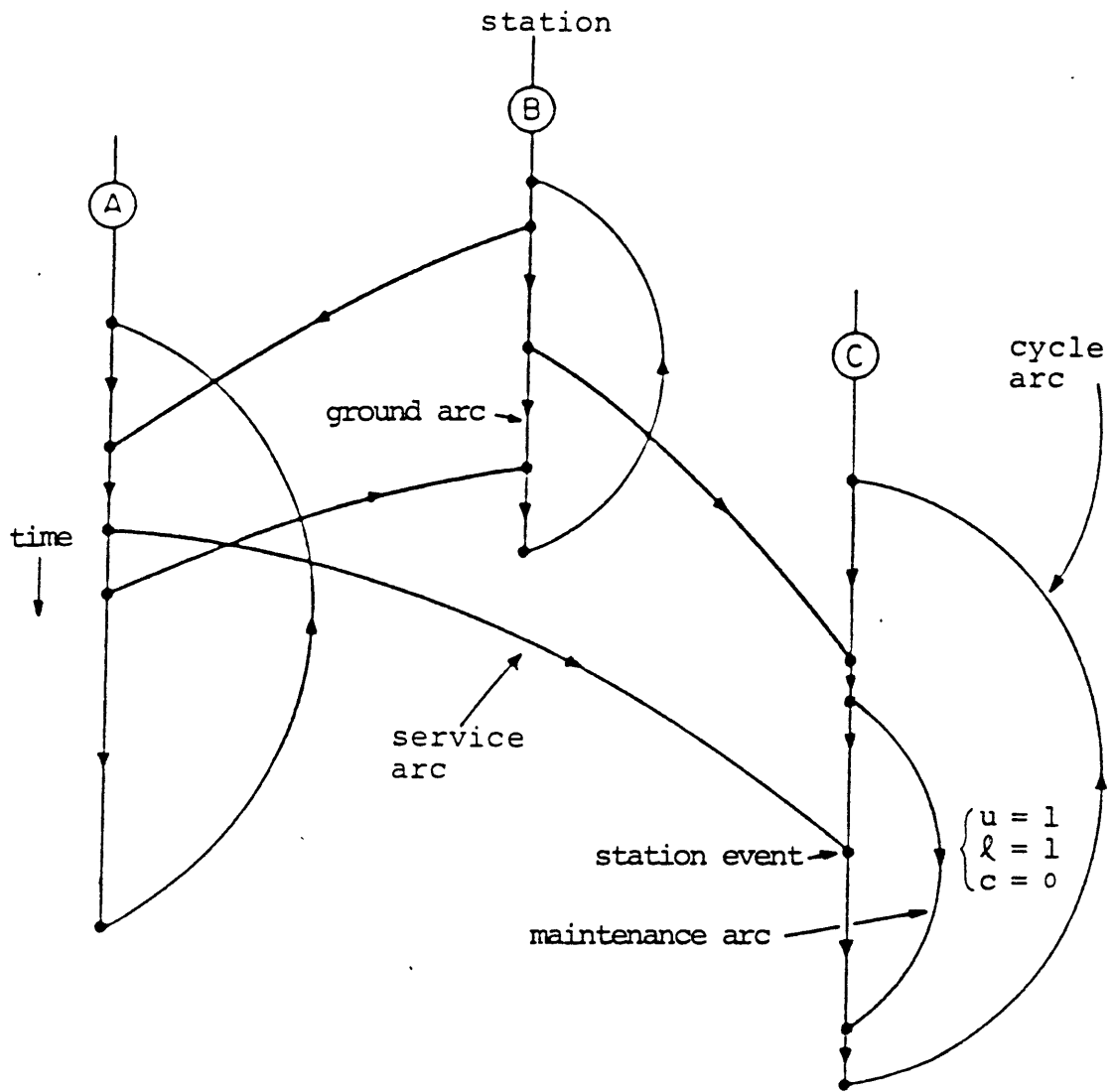


Figure 94 - Schedule chart with maintenance arc

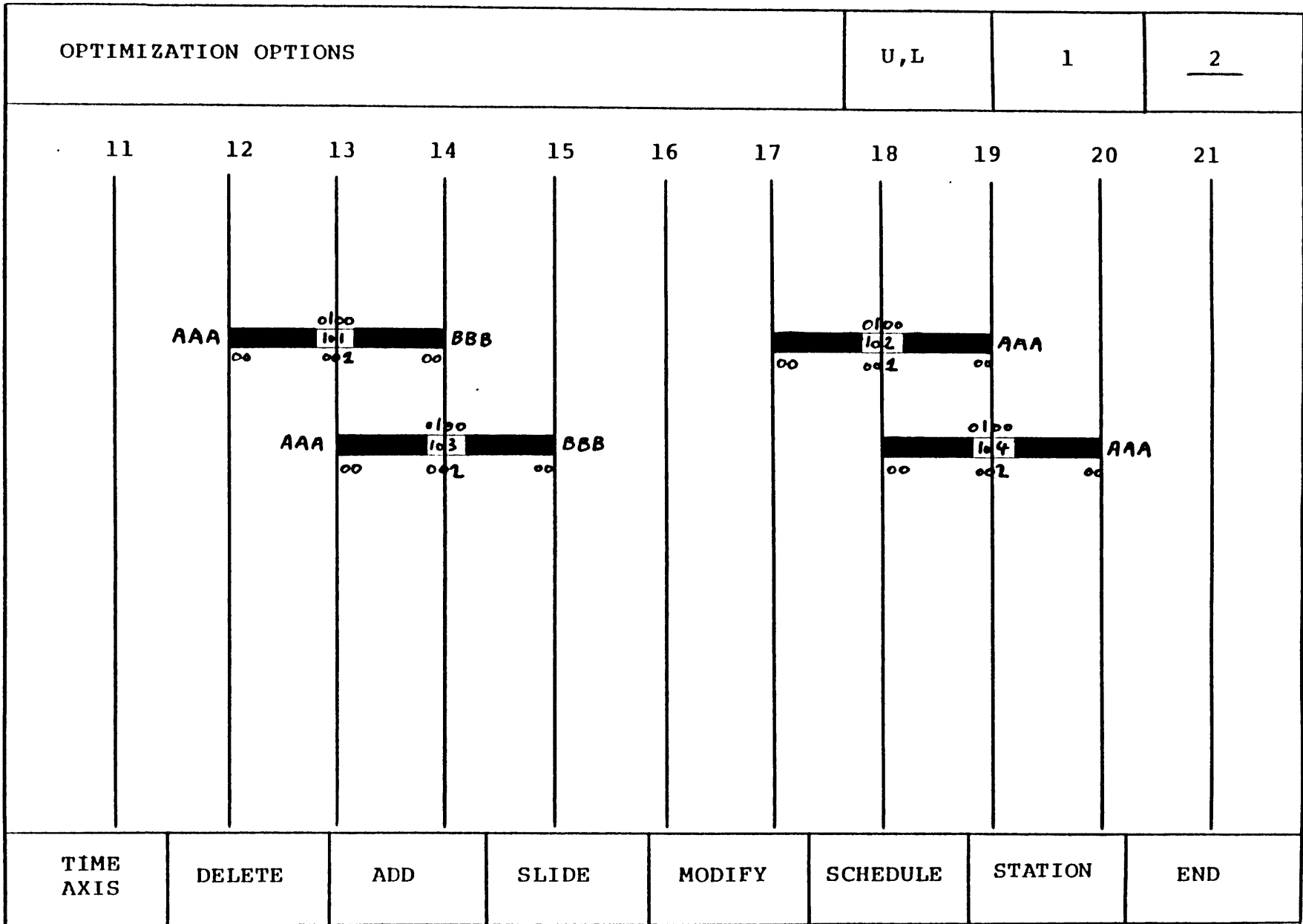


Figure 95 - Schedule plan consisting of two flights

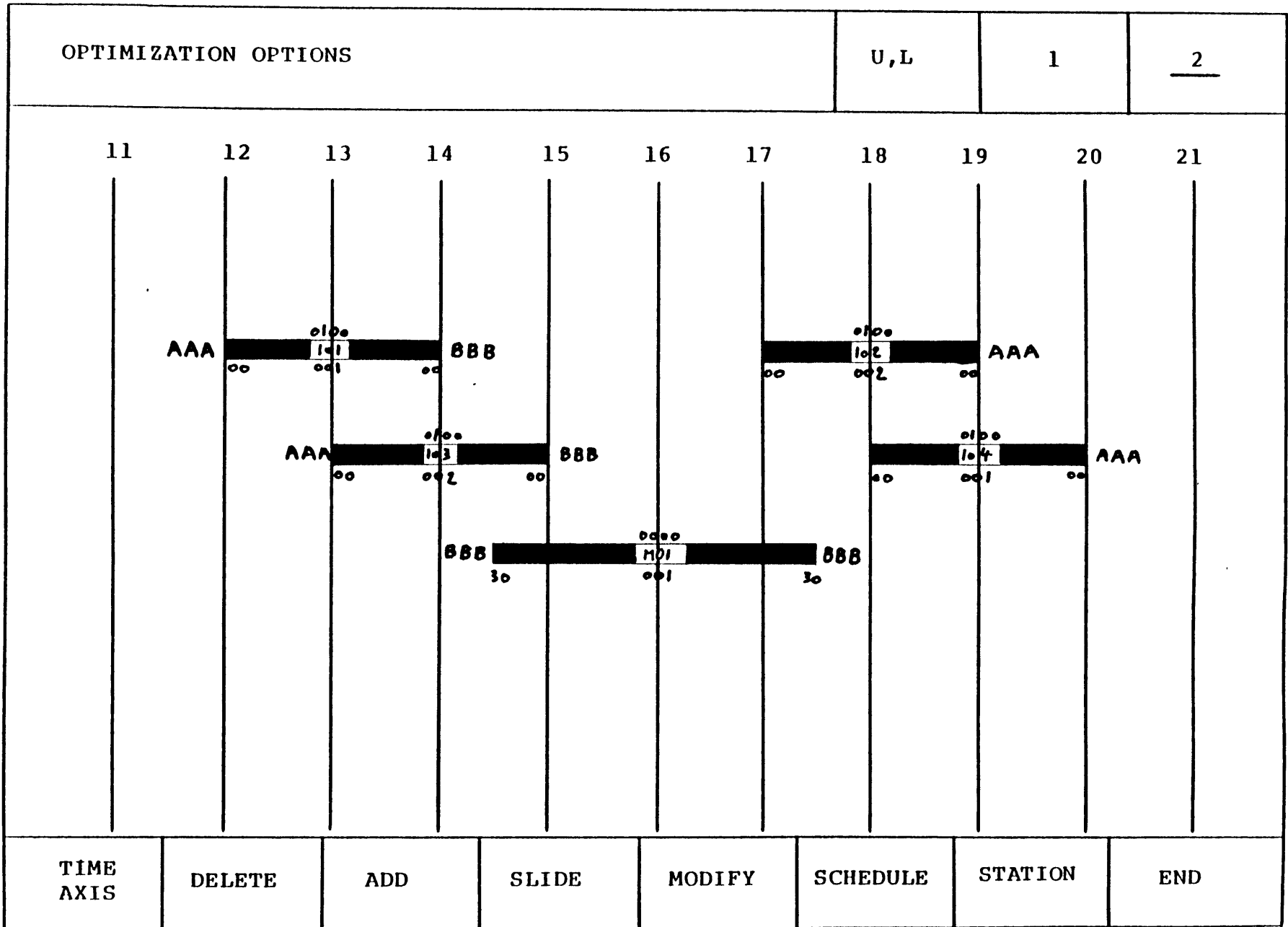


Figure 96 - Aircraft are rerouted after a maintenance stopover has been introduced

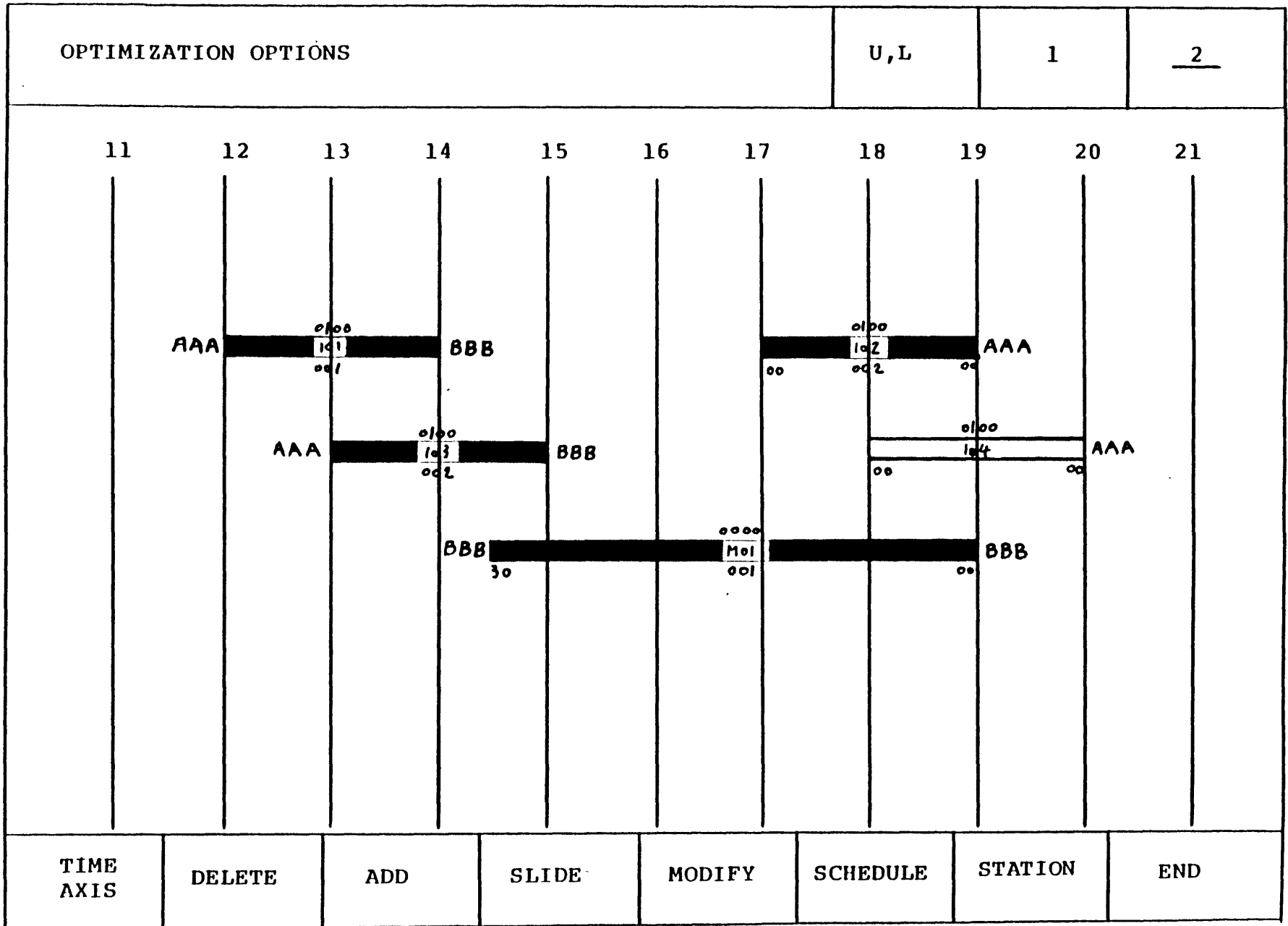


Figure 97 - Loss of service on a flight segment
when maintenance stopover is too long

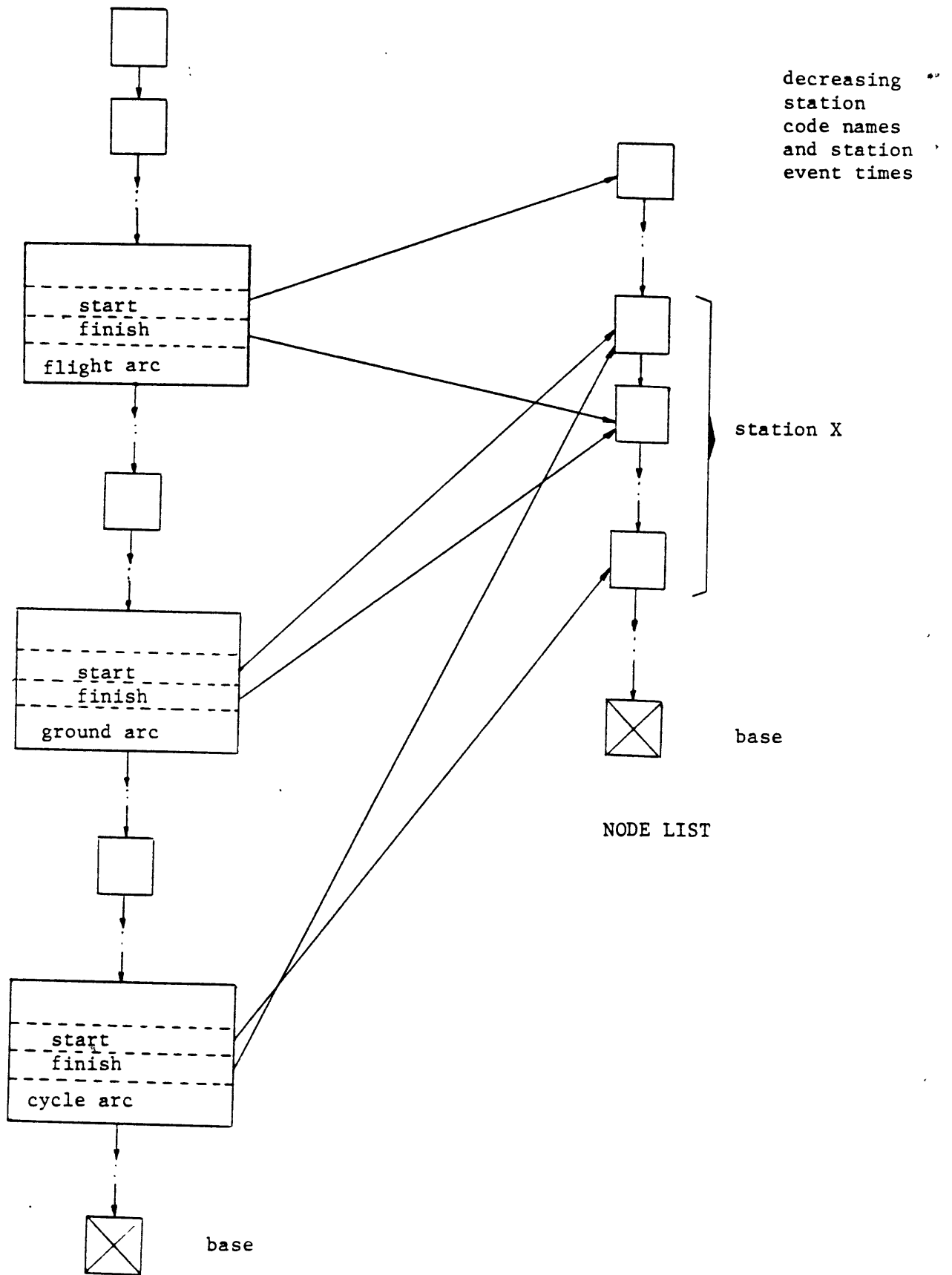


Figure 98 - The network data structure consists of a node linked list and an arc linked list

Appendix B

Data Structure

There is only one network that corresponds to a single type of schedule plan, but there are many data structures that can represent a network. The linked list, being a dynamic data structure, is an obvious choice here considering the demands of a dynamic scheduling environment; it not only conserves memory space, but also allows for rapid access to its records. A node linked list and an arc linked list that are composed of records of the format given below describe the network of a schedule plan:

node: record with

- a pointer to the next record
- the station code name
- the time of arrival or departure
- the day of arrival or departure
- the simplex multiplier
- the first node label: a pointer to the preceding node
in the flow-augmenting path
- the sign of the first node label

- the second node label: the maximum allowable flow in the flow-augmenting path
- a pointer to the arc record of which this node represents one station event

arc: record with

- a pointer to the next arc record
- a pointer to a node containing information on the start of this arc's flight segment
- a pointer to a node containing information on the finish of this arc's flight segment
- the day of occurrence of the flight segment
- the lower bound on the arc flow
- the upper bound on the arc flow
- the flight arc value or the aircraft daily ownership cost
- the arc flow
- the number of the corresponding flight segment
- the arc label: the number of the flight to which this flight segment belongs

The node records are ranked alphabetically and sequentially; one node precedes another in the linked list if it has an alphabetically higher station code name or if it has the same station code name, but a later event time. Ground and cycle arc records, identical to flight segment arc records, can be created and linked to the arc list only after the node list has been created because:

-The start and finish fields of the ground arc records have to point to consecutive event pairs at each station in the node list.

-For the daily schedule plan, the cycle arcs have to point respectively to the last and the first event at each station.

-For the multiple-day schedule plan, the source and sink arcs have to point to all last and first stations events.

The node linked list and arc linked list of the daily schedule plan and the relationship between their records are presented in Figure 98.

Bibliography

- [1] Etschmaier, M.M. and Mathaisel, D.F.X. Aircraft Scheduling: the State of the Art, Proc. of XXIV AGIFORS Symposium, Strasbourg, France, 1984.
- [2] Keen, P.G. and Morton M.S. Decision Support Systems - An Organizational Perspective, Addison-Wesley, Reading, Mass., 1978.
- [3] Sprague, R.H. and Carlson, E.D. Building Effective Decision Support Systems, Prentice-Hall, New Jersey, 1982.
- [4] Ford, L.R. and Fulkerson, D.R. Flows in Networks, Princeton U. Press, New Jersey, 1962.
- [5] Basarara, M.S. and Jarvis, J.J. Linear Programming and Network Flows, Wiley, New York, 1977.
- [6] Lawler, E.L. Combinatorial Optimization: Networks and Matroids, Winston, 1976.
- [7] Magnanti, T.L. and Golden, B.L. Network Optimization, MIT, Cambridge, Mass., 1980.

- [8] Simpson, R.W. Scheduling and Routing Models for Airline Systems,
FTL-report R68-3, MIT, Cambridge, Mass., 1969.