Three Models for Scheduling Space Shuttle Missions and Resources

(NASA-CR-140319) THREE MODELS FOR SCHEDULING SPACE SHUTTLE MISSIONS AND RESOURCES Final Report (International Business Machines Corp.) 34 p HC $3.75 CSCL 22A G3/12
N75-10127
Unclas
53181

Jerrold Rubin
IBM Philadelphia
Scientific Center

Final Report
NAS 9-13939
September 4, 1974

The problem considered in this report is that of scheduling space shuttle missions within specified window and resource constraints. For our purposes, each mission can be thought of as a set of one or more assignments, each assignment requiring one unit of some resource (such as one pilot, one mission specialist, one orbiter, etc). Each such assignment may utilize its resource unit for a period of time prior to and/or subsequent to the mission launch date. The length of those periods may differ for different assignments. A given resource unit either is, or is not, qualified for a given assignment, and this is considered known.

Each mission has a "window" or time interval within which its launch date is to be set, else it cannot go. Some missions are required, others are potential with a given value.

The objective is to schedule all required missions, and as many potential missions as will give maximum value, assuring that qualified resource units are available for the mission assignments.

Three models will be described. The first, model A, will ignore qualifications, selecting misssions of maximum value up to the limits of the available numbers of resources. It will also schedule these missions to time periods within their windows. This will establish a target, the best that can be done if all resource units were qualified for all missions.

The second, model B, assumes mission times are unknown and tries to assign qualified resource units to the mission assignments. Model B can use as its input the results of model A.

In case this cannot be done, it may still be possible to assign qualified resource units to the selected missions, by altering launch times. The third and most complex model, model C, uses as input an intermediate result of Model A (which contains the mission selections and certain other information) and attempts to set both the resource unit assignments and the launch times, so as to achieve a feasible total schedule.

Models A and C rely heavily on the assumptions that many missions resemble one another in their pattern of resource usage. Specifically, two missions are called "resource-equivalent" if the assignments on each can be put in 1-1 correspondence so that, for every corresponding pair

a) The same type of resource (orbiter, pilot, etc.) is required
b) The length of usage prior to launch is the same
c) The length of usage during and subsequent to launch is the same.

Two resource-equivalent missions may have different windows, and different qualification requirements for the resource unit needed.

Models A and C will be easier to solve if there are a small number of resource equivalence classes. In sample data supplied by NASA, there were four such classes among 58 missions.

The number of variables in models A and C depends upon the number of time periods. Since windows extended over one year, a rather gross granularity of one week was used, giving 54 time periods to be considered in the NASA data. The weekly period is applied both to mission windows and to lengths of usage of resource units.

Certain other assumptions and simplifications are discussed at the end of this report.

Model A: Schedule missions within their windows considering overall availability of resources, but not their qualifications.

Set $Y_{ij} = 1$ If mission i launch date is in time period j

$Y_{ij} = 0$ Otherwise

$S_i = 1$ If mission i goes, 0 if not. Of course, if i is a required mission, $S_i = 1$.

Then, $$\sum_j Y_{ij} = S_i \qquad \text{(A.1)}$$

For time period j, resource A, let

N_i^A = number of units of resource A needed by mission i

R_j^A = number of units of resource A available in time period j

d_i^A = length of usage of resource A prior to launch of mission i

e_i^A = length of usage of resource A subsequent to (and including) launch of mission i

(Note: For notational simplicity, we assume all assignments of resource A in mission i have the same values for d_i^A, e_i^A. The model is easily extended if this is not the case.)

In order to state that no more than R_j^A units of resource A can be used in time period j, note that a mission i starting at time $j-e_i^A + 1$ will still be using resource A at time j, and a similar statement holds for time $j + d_i^A$.

Hence,

$$\sum_i N_i^A \sum_{m=j-e_i^A+1}^{m=j+d_i^A} Y_{im} \leq R_j^A \qquad \text{(A.2)}$$

The objective would be to maximize

$$\sum_i V_i S_i \qquad \text{(A.3)}$$

where V_i = value of mission i

Now set

$$X_j^k = \sum_{i \in E_k} Y_{ij} \tag{A.4}$$

where E_k is the k^{th} resource-equivalence class. Then X_j^k represents the number of class k missions launched in time period j.

We can formulate this as a combination of several transportation problems with variable right hand sides.

$$\sum_j Y_{ij} = S_i \qquad i \in E_k \tag{A.5a}$$

$$\sum_{i \in R_k} Y_{ij} = X_j^k \tag{A.5b}$$

and an integer program on the X_j^k variables

$$\sum_k N_k^A \sum_{m=j-e_k^A+1}^{m=j+d_k^A} X_m^k \leq R_j^A \tag{A.6}$$

where N_k^A, e_k^A, d_k^A now all refer to the common values for equivalence class k. There is one such transportation problem for each class k.

Since there are no costs on the variables Y_{ij} of the transportation problems, one must only assure that the latter are feasible. If one could impose further conditions besides (A.6), and the obviously necessary

$$\sum_j X_j^k = \sum_{i \in E_k} S_i \tag{A.7}$$

on the variables X_j^k and S_i so as to guarantee their feasibility it would be possible to solve a problem with the much smaller number of variables X_j^k and S_i, using (A.3) as the objective function, and then, given the values of X_j^k and S_i, proceed to solve the transportation problems knowing that their feasibility is guaranteed.

This, in fact, can be done. The transportation problem variables Y_{ij} have capacity 0 (if j is outside the window of mission i) or infinity (it is not necessary to impose an explicit capacity of one, since S_i is already bounded by one). Appendix 1 shows that the feasibility conditions are of the form

$$\sum_{j\epsilon J} X_j^k \geq \sum_{i\epsilon I} S_i \qquad \text{(A.8)}$$

where J = consecutive set of time periods from some left window boundary to some right window boundary

I = all missions of class k with window entirely contained in J

There will be one equation for each term (l, u) of the cross product L x U, where

L = Set of left window boundaries
U = Set of right window boundaries

except when the corresponding set I is empty.

For more detail, see Appendix 1.

The solution to the System (A.6, A.7, A.8) with objective (A.3) will provide "slots" into which specific missions can be slipped. For example, $X_{10}^3 = 2$ would mean that two missions of resource-equivalence class 3 are to be schedule for launch in the $10^{\underline{th}}$ time period, without specifying which two. The S_i, of course, tell which missions are selected and which are not. This information will be useful to model C, which will assign specific missions, as well as specific qualified resource units, to these "slots". Thus, depending on our purposes, we may or may not proceed to the solution of the transportation problems following the determination of X_j^k and S_i.

Model B: Assignment of qualified resource units to a fixed set of mission assignments at known times.

Notice that we can independently treat each type of resource; hence assume we are dealing with only one type of resource (e.g., mission specialists).

Let $V_{mi} = 1$ if unit m is assigned to mission assignment i, 0 otherwise. (Recall there may be multiple assignments in one mission, each assignment requiring one unit.) If m is not qualified for i, then $V_{mi} = 0$.

$$\sum_{m} V_{mi} = 1 \qquad \text{all } i \qquad (B.1)$$

states that every assignment must be covered. Also,

$$\sum_{i \in I(k)} V_{mi} \leq 1 \qquad \text{all } m, k \qquad (B.2)$$

where I (k) = set of mission assignments that are active when mission k initiates the use of its resource unit (not necessarily the launch time).

Condition (B.2) states that any individual must not be used on two assignments simultaneously.

Note that it is not necessary to have a simultaneity constraint for each time period, since if a unit is simultaneously assigned to a set of two or more missions at any time, it will also be simultaneously assigned at the latest starting time of that set of missions.

If slack variables were added to change equation (B.2) to equalities

$$\sum_{i \in I(k)} V_{mi} + S_{mk} = 1 \qquad (B.2a)$$

then (B.1) and (B.2a) would form a set partitioning problem, which should be readily solvable.

If certain units are equivalent (are qualified for exactly the same missions), we may regard m as an equivalence class, V_{mi} as determining whether any unit of class m is assigned to i, and modify (B.2) to

$$\sum_{i \in I(k)} V_{mi} \leq N_m \qquad \text{all } m, k \qquad (B.2b)$$

where N_m = number of units in class m.

For example, for pilots and commanders, on the sample data, only 6 equivalence classes exist. Thus if, on the average, a man is qualified for half the assignments, then assuming 50 missions, 52 assignments, there will be 156 0-1 variables for commanders and pilots. The objective would be to determine feasibility.

One could similarly take advantage of equivalence of mission assignments with respect to resource units - i.e., two mission assignments would be equivalent if exactly the same units were qualified. Then V_{mi} would tell the number of units of class m assigned to mission assignments of class i, and B.1 would be modified to

$$\sum_m V_{mi} = \text{no. mission assignments in class } i \qquad \text{(B.1a)}$$

Standard branch and bound techniques can be used to solve (B.1a, B.2b).

Model C: Assign missions, times and multiple units of multiple resources. This is the most comprehensive, and the largest model. Consider a transportation problem for each time j, each resource type, and each resource equivalence class k of missions.

Let

r index the mission assignments
i index the missions
$M(r)$ = mission corresponding to assignment r
W_{ij} = 1 iff mission i is launched at time j
X^k_{mj} = 1 iff unit m is assigned to a mission of class k launched at time j
Y_{mrj} = 1 iff unit m is assigned to assignment r, which is launched at time j

$$\begin{array}{r|c|l} \text{mission assignments } r & Y_{mrj} & W_{ij} \\ \hline & X^k_{mj} & \end{array}$$

$$\sum_m Y_{mrj} = W_{ij} \quad \text{where } i = M(r) \tag{C.1}$$

$$\sum_{M(r)\in E_k} Y_{mrj} = X^k_{mj} \tag{C.2}$$

Note that W_{ij} may appear more than once as a right hand side of (C.1). In fact, it will appear exactly N_k times, where

N_k = number of assignments of this resource for a class k mission

We can guarantee the feasibility of this transportation problem by imposing feasibility constraints (See Appendix 2), which are of the form

$$\sum_{i\in I_s} W_{ij} \le \sum_{m\in \bar{S}} X^k, \quad N_k \sum_{i\in E_k} W_{ij} = \sum X^k_{mj} \tag{C.3}$$

where S = a subset of men
I_s = the maximal subset of missions corresponding to S
$\bar{S}$ = complement of S
R_k = resource equivalent class k

It is possible to reduce the problem by considering qualification equivalence classes of men or mission assignments. Two missions may be defined as "qualification-equivalent" if

a) They are in the same resource equivalence class
b) for each pair of corresponding assignments, precisely the same resource units are qualified.

Two resource units may be regarded as equivalent if they are qualified for precisely the same mission assignments.

We can now redefine W_{ij} as the number of missions of class i launched at time j.

If W_{ij} is now to refer to an equivalence class of missions, we must assure that they can be scheduled within their windows. If windows are narrow, the number of non-zero W_{ij} may be acceptable. If windows are broad, we could set up a second transportation problem:

missions in a qualification equivalence class

$$\begin{array}{c|c|c} & \text{time} & \\ \hline & Z_{nj} & S_n \\ \hline & W_{ij} & \end{array}$$

$$\sum_j Z_{nj} = S_n \tag{C.4}$$

$$\sum_{n \epsilon i} Z_{nj} = W_{ij} \tag{C.5}$$

S_n = 1 iff mission n goes.

Z_{nj} = 1 iff mission n is launched at time j.

We must then impose window feasibility conditions as in model A, of the form (see Appendix 1).

$$\sum_{j \epsilon J} W_{ij} \geq \sum_{n \epsilon I} S_n \ , \ \sum_j W_{ij} = \sum_n S_n \tag{C.6}$$

The simultaneity conditions, for an equivalence class m of men, are

$$\sum_k \sum_{j \epsilon T(k,J)} X^k_{mj} \leq N_m \quad \text{each } J \tag{C.7}$$

where N_m = number men in equivalence class m.

T(k,J) = interval of resource usage if mission of class k is launched in time J.

We could also impose a volume restriction on the use of other resources. This would involve stating that no more than N units of each resource are in use at any one time. It would not guarantee that qualified resources are available, but would apply to resources for which qualifications were not a critical consideration.

The condition would be

$$\sum_m \sum_k N_k \sum_{j \epsilon T(k,J)} X^k_{mj} \le N \quad \text{each J, each resource type} \tag{C.8}$$

(Resource equivalence classes must be defined with respect to all resources considered).

For the NASA sample data, there are somewhat fewer than 1000 W_{ij} variables. It may be possible to reduce this figure further by different treatment of missions wherein only one assignment is of a resource for which qualifications play an important role. For example, if it should turn out that it is easy to assign qualified commanders and pilots to any proposed schedule, then we can leave this task to model B, after mission times have been set. In model C only a volume requirement would be needed for commanders and pilots, and qualification considerations would involve only mission specialists. Then any mission requiring only one mission specialist could be treated as follows:

Consider the transportation problem posed by the following tableau:

	man 1	man 2			
missions in class k	Time = 1,52 Y_{mij}	Time = 1,52	- - - - - - - -		S_i
	X^k_{mj}				

$$\sum_m \sum_j Y_{mij} = S_i \tag{C.9}$$

$$\sum_i Y_{mij} = X^k_{mj} \tag{C.10}$$

S_i = 1 if mission i goes, 0 otherwise

X^k_{mj} = 1 if man m goes on a mission at time j, 0 otherwise

Y_{mij} = 1 if man m goes on mission i at time j, 0 otherwise.

Certain Y_{mij} may be forced to be zero, either because j is outside the window of mission i, or because man m is not qualified for mission i. Therefore, the above is a "capacitated" transportation problem, with capacities either zero or infinity.

That a man cannot be used simultaneously on two missions can be stated:

$$\sum_k \sum_{j\ T(k,J)} X^k_{mj} \leq 1 \qquad \text{each } J,m \qquad (C.11)$$

where T(k,J) = time interval around J during which missions of class k are actively using the resource involved.

We could then proceed to impose feasibility constraints, solve the integer program in the X^k_{mj} variables, then solve the transportation problems, as in model A. However, both the number of constraints and the number of variables in either the single-assignment, or the multiple assignment case, can grow rather large. Let us therefore attempt to find another method of solution, other than a straightforward attack on a large integer program.

We have already achieved in model A a target optimum, which represents those missions of greatest value for which resources are available, if their qualifications were not considered. The missions are given by the S_i variables, and "slots" for missions of the different resource equivalence classes are given by the X^k_j variables.

If we were unable to create a schedule to match these slots, because of insufficiently qualified personnel, it would mean that lack of qualifications was having serious impact on scheduling of space shuttle missions. Since the relative cost of qualifying personnel (compared to the cost of a mission itself) is slight, we will make the assumption that qualifications will impact the schedule only to the extent of requiring some shifting of mission times within the slots described above, but not to the extent of requiring different missions to be flown, or a different set of slots i.e., the S_i, X^k_j variables will stay the same. If we are successful at establishing values of X^k_{mj} and W_{ij} so that a feasible solution is established, we know that it will be the optimal solution since we are informed by model A that missions of no greater value can be squeezed in. If not, we may have to attempt to solve model C, with its large number of variables and constraints (several thousand of each) directly.

We can regard the problem as follows:

a) on the one side, for each resource we must schedule units into the preassigned slots, so that

$$\sum_m X^k_{mj} = N_k X^k_j \tag{C.12}$$

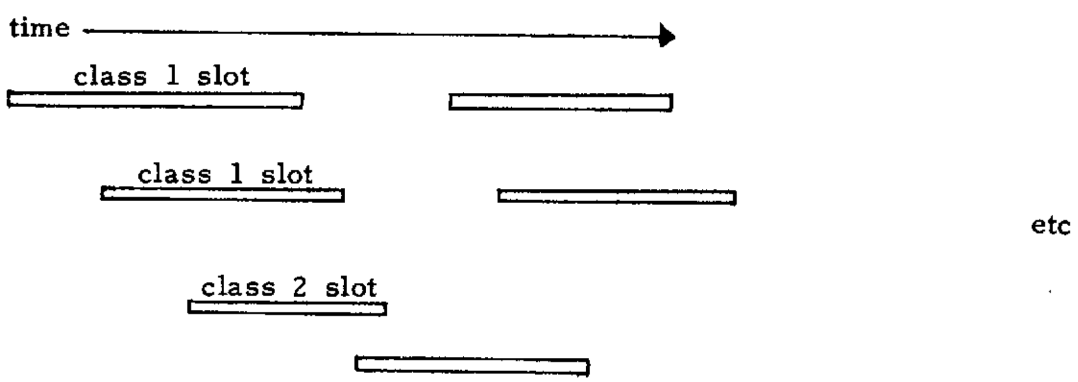
where N_k = number of assignments of the resource for a class k mission

b) on the other, we must schedule missions to the same slots, so that

$$\sum_{i \epsilon E_k} W_{ij} = X^k_j \tag{C.13}$$

Furthermore, these assignments must mesh in the sense that a unit available for a mission assignment must be qualified for it.

Nevertheless, problem (a) above, as an independent problem, is not difficult. Disregarding qualifications, there is a very simple means of establishing appropriate X^k_{mj} variables, and that is to perform a "left-right" assignment. Simply draw a diagram of the slots to be filled, with a time-line running from left to right:



Each slot represents the full usage of the resource unit, including usage prior and subsequent to launch.

Starting with the full pool of resource units, assign one randomly to the first slot (the one extending farthest left), and remove that unit from the pool. Repeat with the next leftmost slot, remembering first to restore to the pool any unit that has again become available. Since at any time period we are guaranteed (by model A) not to use more resources than available, this method will provide a complete assignment of resource units to the slots.

Note that specific missions have not yet been assigned.

Suppose we perform the above process for each resource type. Then we could solve a system in W_{ij} alone, consisting of equations C.3 and C.6 for the multiple unit missions. For missions requiring only one resource unit, only the transportation problem C.8, C.9 need be solved. In both cases, feasibility would be the only objective.

Further, since many values of X^k_{mj} would be selected as zero, the transportation problems, and the feasibility conditions, would reduce sharply in size and number. For example, the problem C.1, C.2 would on the average reduce to a one or two column problem.

However, we must seek justification for this procedure of almost randomly setting some variables (X^k_{mj}) and solving for a feasible selection of the others (W_{ij}). The problem C.9, C.10 will represent roughly a 50 x 50 assignment problem, for the largest resource equivalence class of the NASA sample data. If all men were qualified for all mission assignments, we would be certain of its feasibility, since window feasibility was assured by model A. The question therefore arises, how often will this feasibility be disturbed by imposing a relatively random pattern of qualifications on the capacities.

This problem has not been directly examined. But some experimentation has been done with randomly generated capacitated assignment problems of various sizes. The results of these suggest that, if about 50% of the cells have capacity zero, and the other 50% have capacity 1 a random 50 x 50 problem is almost certain to be solvable, and the chance that a solution exists increases with size. It is precisely where a great deal of flexibility exists, in switching missions around, that solutions are likely to be found, once the X^k_{mj} variables are set.

However, we must also worry about those cases where little flexibility exists i.e., where the resource equivalence class is small, the mission window is tight, or there are few qualified resource units available for a mission assignment. We must assure for these critical missions that X^k_{mj} has been set so that it is at least possible to assign a qualified man to each mission assignment, all at the same time within the mission window.

To meet this need, we adopt a heuristic which orders mission assignments according to a measure of criticalness, based on the above criteria. Then, given a left-right assignment of the X^k_{mj}, which of course satisfies C.12, we start assigning missions to times within their windows, and switching X^k_{mj} accordingly so that, at least for the most critical assignments, qualified men will be available at the right times. After carrying this process to its limit, we obtain a set of values for X^k_{mj} which is more likely than the original set to provide a feasible solution for the system. This process is described in detail in Appendix 3.

Since random factors are involved in the selection of the X^k_{mj} (both in the left-right assignment, and in the switching procedure), if a feasible solution for the W_{ij} is not found, we could repeat the procedure with a new selection of X^k_{mj}. Eventually we might have to give up, and admit that a solution adhering to the results of model A is either difficult or impossible to find. We would then have the unpalatable option of running the complete model C, or performing the scheduling manually.

Note that in spite of the heuristics, any solution found is an optimum, since it adheres to the results of model A, and this was optimum for the case in which all units were qualified for all mission assignments. Trouble would occur only if we cannot find a solution, in which case it is probable that one would regard resource units as insufficiently qualified. That is, qualifications should ideally be kept substantial enough that solutions are easily found.

Sample Data - Assumptions and Simplifications

Runs on models A and B were made with sample data supplied by NASA. This consisted of 58 missions, 50 of which were required missions. Each mission used a commander, a pilot, and either one or two mission specialists. In addition, each must be flown from either the Eastern Test Range (ETR) or the Western Test Range (WTR).

The only distinction, for our purposes, between commanders and pilots is that a pilot must fly two missions before qualifying as a commander. On the assumption that it would be simple to switch assignments if no commander was assigned to some early mission, this condition was ignored, and commanders and pilots were pooled for these tests.

Several other simplications were used, some of which point out shortcomings in the model (though it is not clear how serious these are). Skills will lapse if not used - however, refurbishment of such skills is apparently not difficult, so this was ignored in the scheduling. Similarly, training in new skills is possible, though time consuming, and would have to be decided external to the models. A resource unit engaged in such training could be made unavailable to the model for the training period, and its qualification pattern updated subsequent to such training (note that in model C the qualification pattern can be given independently for each time period).

Although standard maintenance periods can be included as part of the usage of a resource unit subsequent to launch, longer than usual maintenance periods may be periodically required, depending on the amount of usage a resource unit has received. If the approximate date of such maintence can be estimated, it may be possible to treat it as a mission (which requires only that unit, and for which only that unit is qualified).

Vacations for crew members could conceivably be treated similarly. However, we must remember that special purpose missions such as these increase the number of resource equivalence classes, and hence the timing of the computer runs. For example, vacations for commander-pilots, vacations for mission specialists, and maintenance periods for orbiters would add three such equivalence classes. These were not used for the test runs.

Since a weekly grid was chosen, the distinction between three day missions which use resources for 5½ weeks, and seven day missions which use resources for 6 weeks (including one week of rest following the refurbishment period) is blurred, and was ignored in these tests.

Finally, a bound of 3 was placed on the X_j^k variables of model A, in an attempt to avoid too many launches in any one week (both the ETR and WTR can support one launch every two days). It clearly would have been better to introduce the constraint

$$\sum_k X_j^k \leq 3 \qquad \text{for each } j,$$

but this was not done.

Results

Runs were made by using APL to set up the models and the IBM MPSX program to run the integer programs. The standard branch and bound strategies contained in MPSX were used. A network Flow code, in APL, was used to solve the transportation programs.

Runs of model A on MPSX took between 7 seconds and 5 minutes with a surprising variation in time due to small changes in input. The result shown in Table 2 took 30 seconds, used both mission specialists and commander-pilots, but simplified the latter by assuming an available total of 7 and a requirement of 1 for each mission (instead of a total of 15 and a requirement of 2 for each mission). Since there is no variation in the commander-pilot requirement, this is clearly equivalent.

The four transportation problems for model A (one for each resource equivalence class) took a fraction of a second each.

When the result of model A was input to model B to attempt the assignment of qualified mission specialists, an infeasibility resulted, showing that it was impossible to assign qualified men without reassigning mission times. An assignment of commander-pilots was also attempted, but workspace size problems in APL prevented completion of this run.

Workspace size difficulties also prevented running the full data on model C. However, a smaller problem was run on model C with results shown in Table 4. It is notable that with the first attempt at selecting values for X^k_{mj}, no solution for the W_{ij} variables can be found, but at the second try a solution (hence the optimal solution) is found. A more elementary switching procedure than that described in Appendix 3 was used for this run.

WINDOW		R OF P	VALUE	RESOURCE 1 NO. UNITS	RESOURCE 1 PRIOR+SUBS. USAGE		RESOURCE 2 NO. UNITS	RESOURCE 2 PRIOR+SUBS. USAGE	
1	9	1	0	1	2	4	1	2	4
1	36	1	0	1	2	4	1	2	4
1	40	1	0	1	2	4	1	2	4
1	53	1	0	1	2	4	1	2	4
5	9	1	0	1	2	4	1	2	4
9	14	1	0	1	2	4	1	2	4
9	14	1	0	1	2	4	1	2	4
9	23	1	0	1	2	4	1	2	4
9	27	1	0	1	2	4	1	2	4
9	27	1	0	1	2	4	1	2	4
9	40	1	0	1	2	4	1	2	4
9	40	1	0	1	2	4	1	2	4
14	27	1	0	1	2	4	1	2	4
14	31	1	0	1	2	4	1	2	4
14	44	1	0	1	2	4	1	2	4
14	49	1	0	1	2	4	2	2	4
14	53	1	0	1	2	4	1	2	4
18	23	1	0	1	2	7	1	2	7
23	36	1	0	1	2	4	1	2	4
27	44	1	0	1	2	4	1	2	4
27	49	1	0	1	2	4	1	2	4
27	53	1	0	1	2	4	1	2	4
31	36	1	0	1	2	4	1	2	4
31	40	1	0	1	2	4	1	2	4
31	40	1	0	1	2	4	2	2	4
31	40	1	0	1	2	4	1	2	4
31	44	1	0	1	2	4	1	2	4
31	53	1	0	1	2	4	1	2	4
36	49	1	0	1	2	4	1	2	4
36	53	1	0	1	2	4	1	2	4
40	53	1	0	1	2	4	1	2	4
44	49	1	0	1	2	4	1	2	4
44	49	1	0	1	2	4	1	2	4
44	49	1	0	1	2	4	1	2	4
49	53	1	0	1	2	4	1	2	4
49	53	1	0	1	2	4	2	2	4
1	9	1	0	1	2	4	1	2	4
5	14	1	0	1	2	4	1	2	4
5	40	1	0	1	2	4	1	2	4
18	23	1	0	1	2	4	1	2	4
18	23	1	0	1	2	4	1	2	4
18	31	1	0	1	2	4	1	2	4
23	44	1	0	1	2	4	1	2	4
27	36	1	0	1	2	4	1	2	4
27	49	1	0	1	2	4	1	2	4
36	53	1	0	1	2	4	1	2	4
40	53	1	0	1	2	4	2	2	4
49	53	1	0	1	2	4	1	2	4
22	36	0	2	1	2	4	2	2	4
15	54	0	2	1	2	4	1	2	4
23	31	0	1	1	2	4	2	2	4
1	53	0	1	1	2	4	1	2	4
43	46	0	1	1	2	4	2	2	4
25	26	0	1	1	2	4	1	2	4
1	27	0	1	1	2	4	1	2	4
44	53	0	1	1	2	7	2	2	7
46	48	0	1	1	2	4	1	2	4
29	53	0	1	1	2	4	1	2	4

Input for runs on model A

1. WINDOW is in weeks, inclusive
2. R = 1 (required mission)
   P = 0 (potential mission)
3. SUBSEQUENT USAGE OF A RESOURCE UNIT INCLUDES WEEK OF LAUNCH

	AVAILABLE UNITS
RESOURCE 1	8
RESOURCE 2	7

TABLE 1

Mission	Time	Mission	Time
1	2	31	41
2	2	32	44
3	3	33	45
4	3	34	45
5	8	35	50
6	9	36	53
7	9	37	4
8	10	38	8
9	10	39	20
10	10	40	21
11	14	41	22
12	14	42	26
13	15	43	39
14	15	44	34
15	16	45	41
16	22	46	41
17	16	47	47
18	18	48	51
19	23	49	35
20	27	50	54
21	27	51	29
22	28	52	4
23	31	53	45
24	32	54	26
25	38	55	4
26	32	56	51
27	33	57	47
28	33	58	51
29	39		
30	39		

Results of model A scheduling 58 missions, some with 2 mission specialists - all missions are scheduled.

Table 2

INPUT TO MODEL C

XMAIS From model A

```
1 1
5 1
7 2
```

implies $X_1^1 = 1$, $X_1^2 = 1$, $X_3^2 = 2$

SMARS

```
1 1
3 1
```

implies 1st and 3rd potential missions (missions 2 and 4) are chosen; Mission 3 is not.

U

```
1  2  1  0  1  1  1  0  1  1
1  4  0  5  2  1  1  1  1  1
1  1  0  0  1  0  0  2  0  0
3  4  0 10  2  1  1  1  1  1
2  3  1  0  2  1  1  1  1  1
```

same format as for model A

AV # units of each resource available

```
4 3
```

MR consider both resources 1 and 2

```
1 2
```

T time span weeks 1-4

```
1 2 3 4
```

C prototype missions for each equivalence class

```
1 2 3
```

EQ

```
1 0 0 0 0
0 1 0 1 1
0 0 1 0 0
0 1 0 1 1
0 1 0 1 1
```

pattern of equivalences of one mission to another

MSKILLS1

```
1 1 1
1 1 0
0 1 1
1 0 1
```

resource 1 - 3 skills for each of 4 units

MSKILLS2

```
1 1
0 1
1 0
```

resource 2 - 2 skills for each of 3 units

ASKILLS1

```
1 0 1
0 0 0
0 1 1
1 0 0
0 1 0
1 0 1
0 1 0
1 0 1
```

resource 1 skill requirements
- 8 assignments, 3 skills
(4th assignment is for a mission not selected)

ASKILLS2

```
1 0
1 0
0 0
0 1
0 1
```

resource 2 skill requirements
- 5 assignments, 2 skills
(2nd and 3rd assignments are for a mission not selected)

TABLE 3

```
INITIAL DIMN EU IS1 2
DIMN EU AFTER EMPTY REDUCE IS1 2
    INITIAL DIMN EU IS4 2
    DIMN EU AFTER EMPTY REDUCE IS3 2
    LIST
     1 1 1 0
     1 3 5 0
     1 4 7 0
     2 4 3 0
     2 3 2 0
     1 4 6 0
     1 3 4 0
     1 2 3 0
     2 2 1 0
     1 2 2 0
    SWITCHING, ASST 1K,A IS
    1 2 1 1 1 2 2 1 1 1 2
    XMJKL IS
     1 0
     0 0
     0 1
     0 0

     0 1
     0 0
     0 1
     0 0

     0 0
     0 0
     0 1
     0 0

     0 1
     0 0
     0 1
     0 0
    BMJKL IS
     1 0
     0 0
     0 1
     0 0

     0 1
     0 0
```

RUN OF MODEL C

list of mission assignments arranged according to degree of criticalness

successful switch of X^{k}_{m1}

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

TABLE 4 (1)

```
 0 0

 0 0
 0 0
 0 1
 0 0

 0 1
 0 0
 0 1
 0 0
CANT ASSIGN QUALIFIED MAN TO ASGT 5
CANT ASSIGN QUALIFIED MAN TO ASGT 6
[illegible]1
 1 0
 0 0
 0 1
 0 0

 0 1
 0 0
 0 1
 0 0

 0 0
 0 0
 0 1
 0 0

 0 1
 0 0
 0 1
 0 0
[illegible]2
 0 0
 0 0
 0 0
 0 0

 0 0
 0 0
 0 1
 0 0

 0 1
 0 0
 0 1
 0 0
TIME= 0.0666666667 SEC
PROBLEM SOLVED
TIME=0.01666666667 SEC.
TI=0 0
ITOT=1
OBJ. FN. =0
DIFF MAS3 IS3 6
DIFF MAS1 IS1 3
DIFF MAS0 IS0 4
DIFF MAS0 IS0 2
DIFF MAS15 IS4 8
DIFF MAS3 IS2 4
DIFF MAS0 IS0 4
DIFF MAS0 IS0 2
VALUE ERROR
[illegible] CONTROL [illegible] 0 1 +PROGRAM
```

cannot make ad hoc assignments of qualified men

solve single-unit transportation problem

TABLE 4 (2)

[illegible]

TAPED 16.42.58 07/01/74

-10

SOLUTION NOT FOUND

cannot solve integer program for multiple-unit missions with this selection of X^k_{mj}

LIST

```
 1 1 1 0
 1 3 5 0
 1 4 7 0
 2 4 3 0
 2 3 2 0
 1 4 6 0
 1 3 4 0
 1 2 3 0
 2 2 1 0
 1 2 2 0
```

critical assignment list

SWITCHING, ASST 5X,4 IS

2 3 3 1 1 2

switching X^k_{mj} (2nd resource)

XNMJ2 IS

```
 0 0
 0 0
 0 1
 0 0

 0 0
 0 0
 0 1
 0 0

 0 1
 0 0
 0 0
 0 0
```

TNMJ2 IS

```
 0 0
 0 0
 0 1
 0 0

 0 0
 0 0
 0 0
 0 0
     0
 0 1
 0 0
 0 0
 0 0
```

XNMJ1

final values of X^k_{mj} for each resource

```
 1 0
 0 0
 0 1
 0 0

 0 1
 0 0
 0 1
 0 0

 0 1
 0 0
 0 1
 0 0

 0 0
```


TABLE 4 (3)

```
  0 0
  0 0
  0 1
  0 0

  0 0
  0 0
  0 1
  0 0

  0 1
  0 0
  0 0
  0 0
TIME= 0.0666666667 SEC
PROBLEM SOLVED
TIME=0.01666666667 SEC.
PI=0 0
FLOW=1
OBJ. FN. =0
TIME WAS3 IS2 6
TIME WAS1 IS1 3
TIME WAS0 IS0 4
TIME WAS0 IS0 2
TIME WAS15 IS4 8
TIME WAS3 IS1 4
TIME WAS0 IS0 4
TIME WAS0 IS0 2
```

solve transportation problem for single-unit missions

```
NAME      EPTEST

SECTION:  SOLUTION
FUNCTION  --            .00000
STATUS    --  OPTM

SECTION:  CSECTION
  NAME              ACTIVITY
  C......1           1.00000
  C......3           2.00000
SYNTAX ERROR
RUNTRC[21] XVARS←(,1 RECONVERT M/USERCNT),[1.5] M/,USERCMAT
                   ^
      )COPY M 1 RECONVERT
SAVED  13.42.53 07/01/74.
      →21
TIME= 0.1 SEC
PROBLEM SOLVED
TIME=0.01666666667 SEC.
PI=0 0 0 0 0
FLOW=1 0 0 0
OBJ. FN. =0
TIME= 0.1 SEC
PROBLEM SOLVED
TIME=0.01666666667 SEC.
PI=0 0 0 0 0 0
FLOW=0 0 1 0 0 0 1 0
OBJ. FN. =0
TIME= 0.08333333333 SEC
PROBLEM SOLVED
TIME=0.016
```

successful solution of integer program for multiple-unit missions

solve transportation problems for each time period, each resource type, each resource equivalence class

TABLE 4 (4)

PROBLEM SOLVED

PROBLEM SOLVED

PROBLEM SOLVED

ASSIGNMENTS

MISSION	TIME	RESOURCE	UNIT EQ CLASS	ASST NO.
1	1	1	1	1
2	1	1	2	1
2	1	1	3	2
2	1	2	3	1
4	3	1	2	1
4	3	1	3	2
4	3	2	1	1
5	3	1	1	1
5	3	1	4	2
5	3	2	2	1

TABLE 4 (5)

The Minimum Resource Problem

Models A and C deal basically with the problem of maximizing mission value given fixed resources, if mission times are permitted to vary. For fixed mission times, the problem becomes much simpler, since a left-right assignment of resources will not only assign resource units, but also tell us how many unused ones remain. Indeed in this case, the minimum number of resource units needed is precisely the maximum used at any point in time.

The problem of minimizing resource utilization for a fixed number of missions, when mission times are permitted to vary, deserves some comment. This could be done using model A, where all missions are treated as required missions, and R_j^A (the number of units of resource type A available in time period j) is made variable. Costs could be assigned to each R_j^A, and the model minimized on these costs. Probably, we would make R_j^A equal for each j, and with costs of 1 assigned, the problem would involve minimizing the number of resource units used.

If qualifications are involved, it becomes unclear as to what is meant by minimizing resource utilization, since this can probably best be done by more thorough training.

Appendix 1

Conditions on Feasibility of the Mission - Time Period Transportation Problem

Theorem

Given a transportation problem with capacities C_{ij} which are either zero or infinite, and the infinite capacities contiguous in each row, as below:

$$
\begin{array}{ccc}
 & j \longrightarrow & \\
i \downarrow & \begin{bmatrix} 0 & 0 & 0 & C & C & C & C & 0 & 0 & 0 \\ 0 & C & C & C & C & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & \end{bmatrix} & a_i \\
 & b_j &
\end{array}
$$

$$C = \infty$$

$$\sum_{\text{all } i} a_i = \sum_{\text{all } j} b_j \qquad (1)$$

$$\sum_j X_{ij} = a_i \qquad (2)$$

$$\sum_i X_{ij} = b_j \qquad (3)$$

The interval of j for a contiguous row of C's is termed a window.

Let L = set of indexes j that initiate a window.
U = set of indexes j that terminate a window
J_k = the interval (l,u) where the pair (l,u) is the kth element of L x U.
$I_k = \{ i \mid C_{ij} = 0,\ j \in J_k \}$

To prove: The conditions for existence of a feasible solution are:

$$\sum_{j \in J_k} b_i \geq \sum_{i \in I_k} a_i \qquad (4)$$

where there is a constraint for each unique pair (l,u) in LxU such that there is at least one mission with window entirely contained in the interval (l, u).

We assume that the first and last columns each contain an infinite capacity (any column with all zero capacity can be dropped). Assume the inequalities (4) are satisfied. By a known theorem (see "Flows in Networks" Ford-Fulkerson, Princeton U. Press, p. 38) the problem is feasible if for every subset I of the rows, and every subset J of the columns

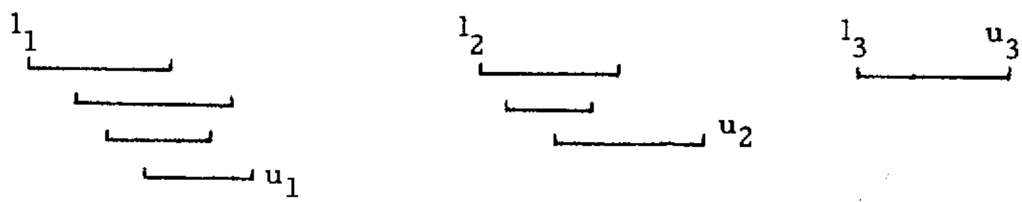
$$\sum_{\substack{i \in I \\ j \in J}} C_{ij} \geq \sum_{i \in I} a_i + \sum_{j \in J} b_i - n \tag{5}$$

where $$n = \sum_{\text{all } i} a_i = \sum_{\text{all } j} b_j \tag{6}$$

This condition is automatically satisfied for any sets I, J if there exist $i \in I$ and $j \in J$ with $C_{ij} = \infty$. Further, for a fixed J_o let $I = \{ i | C_{ij} = 0, j \in J_o \}$. Then let $J = \{ j | C_{ij} = 0, i \in I \}$. Such sets I, J are termed maximal sets since the addition of any other row to I or column to J would introduce some C_{ij} with value ∞. If condition (5) is true for maximal sets I, J, it is also clearly true for any subset $I' \subset I$ or $J' \subset J$. Hence, we need only examine the conditions for maximal subsets.

Suppose now I is a maximal subset of rows. Its union of windows will consist of intervals

$$J_k = (l_k, u_k) \qquad l_k \in L, \quad u_k \in U, \quad l_k < u_k < l_{k+1}$$



for which each corresponding column contains at least one infinite capacity cell; and intermediate intervals

$$J_k^* = (u_k + 1, l_{k+1} - 1)$$

for which corresponding columns have zero capacity. (Note: the initial and final intervals may be of either type.) The maximal subset J* corresponding to I is clearly $\bigcup_k J_k^*$.

Let $J = \bigcup_k J_k$

Let I_k = rows whose windows are in (l_k, u_k).

$I_k \subsetneq I$, since otherwise I would not be a maximal subset, and in fact $I = \bigcup_k I_k$.
Further $I_k \wedge I_j = \emptyset$ if $k \neq j$.

Assuming equation (4) is satisfied,

$$\sum_{j \in (l_k,\, u_k)} b_j \geq \sum_{i \in I_k} a_i \qquad \text{for each } k$$

Adding these equations for each k produces

$$\sum_{j \in J} b_j \geq \sum_{i \in I} a_i$$

Since $\sum_{j \in J} b_j + \sum_{j \in J^*} b_j = n$, and since $C_{ij} = 0$ if $i \in I$, $j \in J^*$,
condition (5) is an immediate consequence, and this establishes feasibility.

The converse, that feasibility (which is equivalent to conditions 5) implies the inequalities (4), is clear.

Appendix 2

Conditions on Feasibility of the Man-Mission Qualification Transportation Problem for a Single Time Period.

In any time period, we have the transportation problem

$$
\begin{array}{l|c|c}
 & \text{mission assignments } j & \\
\hline
\text{resource units } i & X_{ij} & a_i \\
\hline
 & b_j &
\end{array}
$$

(m rows, n columns)

$$\sum_j X_{ij} = a_i \qquad \sum_i X_{ij} = b_j$$

$X_{ij} = 1$ iff unit i takes assignment j

where C_{ij} = capacity = 0 if unit i is not qualified for mission assignment j, infinity otherwise.

a_i = 1 iff unit i is assigned in the time period, (i.e., is assigned to a launch)
0 otherwise

b_j = 1 iff the mission is launched in the time period
0 otherwise

Clearly, we must impose $\sum_{\text{all } i} a_i = \sum_{\text{all } j} b_j$

From appendix 1, we know we need examine only maximal subsets I and J. Each of the 2^m row subsets has a corresponding maximal column subset, hence there are at most 2^m constraints. They are of the form

$$\sum_{j \in J} b_i \geq \sum_{i \in I} a_i$$

where I is a selection of resource units, and J consists of all missions for which some unit in I is qualified.

If J is empty, we can delete the constraint. Also, if I_1, I_2 both have J as the corresponding maximal subset of columns, and if $I_1 \subset I_2$, we can delete the constraint for I_1, J.

We can extend this to equivalence classes of resource units, where all units in a class are equally qualified for the missions. Then we must redefine a_i = no. of units in equivalence class i assigned to a launch in the time period (upper bound is cardinality of the equivalence class).

Appendix 3

Techniques for Switching Assignments of X^k_{mj} To Assure That Critical Missions Can Be Covered

In order to assure that the integer program in the W_{ij} variables, in model C, is feasible, one would like to guarantee that selections of X^k_{mj} are made so that it is at least possible to assign launch dates of critical missions to time periods and have qualified resource units available for the mission assignments. For example, if a mission has window 9-13, and the only resource unit qualified for one of its assignments has been positioned at time periods 8 and 14, it will clearly be impossible to provide an overall feasible solution.

For this purpose, an ad hoc ordering and switching technique has been applied, following the left-right assignment of the X^k_{mj} variables. Recall that a left-right assignment puts men into prestablished "slots", so that the relation

$$\sum_m X^k_{mj} = N_k X^k_j$$

is maintained. During the switching process, we still maintain this equation.

The first step involves ordering all mission assignments, for all resource types, according to the following criteria:

a) percentage of qualified resource units divided by length of use of the unit

b) length of mission window

c) number of missions in the resource equivalence class

That is, assignments of equal value on criterion (a) are ordered according to (b), those equal on (a) and (b) are ordered according to (c), and those equal on all three are ordered randomly. Those assignments of smallest value are listed first, as being the most critical ones.

It is by no means clear that this method of ranking assignments is best, but at least the most critical assignments will appear early.

Note that the size of the resource equivalence class is of importance because there is less flexibility in solving a small transportation problem than a large one. If the transportation problem consists of only one or two mission assignments, for example, we should be particularly careful in assuring that the selection of X^k_{mj} for that class k puts qualified men at the right time periods.

For the purposes of this ad hoc procedure only, tentative launch dates will be set and tentative resource unit assignments made as we examine each critical mission assignment. These launch dates and unit assignments will be discarded prior to execution of the integer program, leaving it the full flexibility of assigning launch dates and resource units.

We examine each assignment once, in order, executing the following steps:

1) If a mission launch date has not yet been tentatively chosen, use as possibilities those dates within the window for which the X_j^k variables have not already been filled. For example, if $X_2^1 = 3$, and three class 1 missions have already been given tentative launch dates in time period 2, we may not use time period 2 as a launch date for another class 1 mission.

   If a mission launch date has been tentatively chosen (i.e., because another assignment of the same mission was more critical and hence appeared higher on the list), use it as the only possibility.

   If the list of possible launch dates is empty, proceed to the next most critical assignment.

2) List all possible values of X_{mj}^k for the resource type involved, such that j is within the list of possible launch dates, k is the correct resource equivalence class for the mission, and m is both qualified for the assignment, and has not yet been tentatively assigned to an assignment higher on the list.

3) If this list is non-empty (i.e., if not all those X_{mj}^k are zero), select one at random, set the tentative mission time, tentatively assign m to this assignment, and proceed to the next most critical assignment.

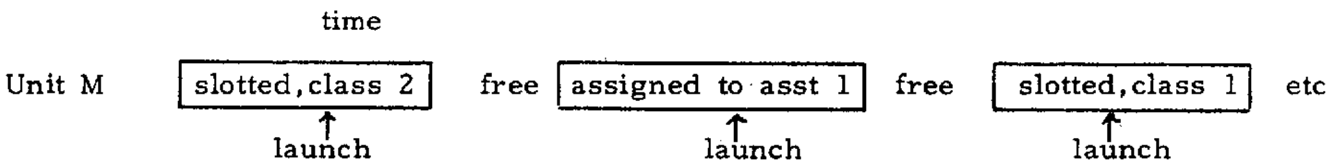
4) If the list constructed in step 2 is empty, we must try to reassign the X_{mj}^k variables so that a qualified resource unit is available at one of the possible launch dates.

We know that an unqualified resource unit must be available, since we have been maintaining the relationship

$$\sum_m X_{mj}^k = N_k X_j^k$$

Let M be such an (unqualified) unit.

We can regard the X^{k}_{mj} variables as providing a time-line for resource unit M - i.e., unit M is "slotted" for missions of class k at time j. The class tells us the length of usage of the unit. At any point during this process, therefore, a unit M is either not being used, slotted but not explicitly assigned, or tentatively assigned to a particular assignment.



In order to switch a qualified unit so that it will take a slot that unit M is currently taking, we must ensure that the result does not involve either unit being simultaneously used by two missions. We can do this by switching an entire "segment" i.e., an interval between two time periods at which both units are free (or are just starting a mission).

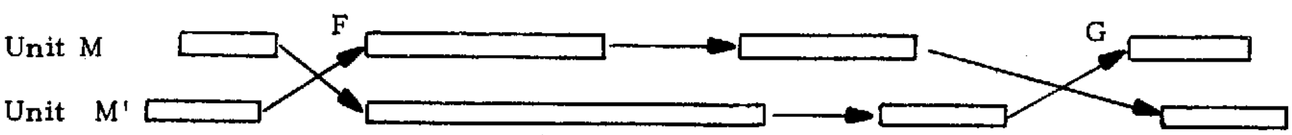


Figure A

In figure A, to avoid simultaneity problems, the segment from F to G must be switched.

Note that a segment is defined based on two resource units, not one. The segments of M with a different unit M" might be entirely different.

Of course, if unit M' has already been explicitly assigned within the segment, we must assure that M is qualified for those assignments, otherwise we cannot make the switch. The same holds for any tentative assignments already made for M.

Thus, the procedure for step 4 involves listing each slot of the correct class, for any available resource unit (even though unqualified). Then for each such slot and for each qualified unit, attempting to switch the smallest segment containing the slot (the smallest segment could conceivably be the entire time-line for the two units).

If any legal switches result, pick one at random, reset X^{k}_{mj} accordingly, and go back to step 2 with the confidence that there is now at least one qualified unit available. If there are no legal switches possible, proceed to the next most critical assignment.

Note that there are two reasons why critical assignments may fail during this process. The first is that, since we are not solving the window transportation problem (which is guaranteed feasible by our choice of X_j^k in model A), but rather are assigning mission launch times in a simple sequential fashion, we may not be able to complete the assignment of launch times within the strictures imposed by the X_j^k variables. Secondly, we may find cases where a simple switch to reposition a qualified unit may not exist. Indeed, if we could complete this ad hoc procedure without either of these problems occuring, we would have in hand a complete and optimal solution to the problem. This is unlikely, however, and it should be kept in mind that the entire purpose of this rather complicated heuristic is simply to find a sensible selection of the X_{mj}^k variables so that it will be likely that the remaining integer program on the W_{ij} has a feasible solution (or, in the case of single-unit missions, the transportation problem).

If the integer program is infeasible then, since random choices were made both in the original left-right assignment of X_{mj}^k, and in the above procedure, one alternative would be to repeat the selection of X_{mj}^k, hoping that the new selection will allow a feasible solution to occur. A few such attempts might be appropriate.

Acknowledgments

I would like to express my deep gratitude to Dr. Alan Hoffman of the IBM T. J. Watson Research Center for guidance in all phases of the project. In particular, Dr. Hoffman pointed out the existence of the theorem giving conditions for the feasibility of a transportation problem, and was the first to recognize that these conditions reduce to a relatively manageable number for this problem.

Also, to Dr. William White for numerous helpful suggestions and discussions, and for his patience in listening to and criticizing all the false steps and wrong paths, and to Dr. Kurt Spielberg for his expertise on integer programming and his critical comments, I express my thanks.

In particular, Dr. White supplied an APL network Flow code and the APL/MPSX interface, both used in the testing of these techniques.