Automated Ship Routing

A.J. Klapp*
Fleet Numerical Weather Central, Monterey, Calif.

The U.S. Navy's Fleet Numerical Weather Central has developed a numerical model to compute realistic time tracks for transoceanic shipping using short-range dynamic forecasts (40-72 hrs) and extended-range statistical forecasts to destination. This model resides in a computer program that is utilized by watch standers to prepare original routes and underway diversions for about 600 U.S. Navy ships a year. Global minimum time routes (GMTR) are computed twice daily for ships using optimum track ship routing (OTSR) service by this program, which considers navigation, environmental surveillance, environmental constraints, and ship performance curves. Maintenance of environmental surveillance and application of environmental constraints is accomplished through grids that represent the oceanic areas of the world and contain environmental parameters such as significant wave height and direction, frequency, fog, currents, and wind; however, any parameter or combination of parameters which can be represented numerically may be used in determining the GMTR. Comparison of the GMTR and the actual route sailed is possible by the computation of a postvoyage verification route using actual environmental data. Ice areas are depicted (through the use of satellite data) and are utilized to constrain the global minimum time route selections. This is a description of the logic and the environmental parameters that produce the automated navigation, route surveillance, original route recommendations, and route verification.

I. Enviromental Arrays

THE first step in routing ships for weather is to store the desired environmental files so that they may be accessed easily by the main routing program. To be able to access these files rapidly, they must be random access files. This is very important because, in order to economize on computer central memory, these files must be read in many times. Once these parameters are stored, they are accessed readily by subroutines that convert latitude and longitude positions to grid positions and then interpolate for time and space. Figure 1 is an outline of the environmental files that are used in ship routing. Wave height is predicted on the basis of wind predictions that drive the spectral ocean wave model (SOWM). This model provides both for growth of waves and propagation of wave energy to grid points that are about 140 miles apart. The wind fog and tropical data are used to avoid areas where there is a high probability of fog or areas where high wind velocities are forecast. The analog wave data are utilized in routing the ships on days 4 through 10 and are a historical sea height forecast based on a best-weather pattern match for 28 yr of history. After day 10, the monthly wave climatology is used to route the ship to its destination, and the monthly currents are used to adjust the ship's speed along the entire route.

II. Automated Navigation

In order for the correct environmental parameters to be selected, the correct position of the ship relative to run time must be determined. From this run-time position, the ship's speed of advance is controlled by a speed profile, which can be updated by speed changes, and the ship's position may be corrected with position reports. This speed profile contains all of the speed changes that the ship is expected to make during its voyage. In the navigation routine, 16 speed changes are possible, and the ship actually can be brought to a full stop. This is done by giving the computer a very slow speed with which to navigate the ship, such as 0.001 knots. The only two limits to the navigation routines are 1) they do not like to operate with a speed exactly equal to zero, and 2) they will not navigate directly over the north or south pole (90N, 90S). Otherwise, they will take any route with less than 30 route points in any hemisphere.

The navigation logic is broken into two sections. The first section determines the ship's correct position at the run time of the computer program. The second section navigates the ship from the run-time position along a track to the ship's destination. In Fig. 2, the ship is in predeparture status. The ship's base speed corrected for wave height by speed reduction curves is used up to 72 h past the run-time dead-reckoning (DR) position, and from there the analog speed reduction factor is used to navigate the ship to its destination.

After the ship has departed, if there is no position report, the ship is navigated up to run time by using the speed that was saved from the previous run. This is the speed that is obtained by taking the distance between the 12-h position of the ship and the departure port and dividing it by the difference in hours between the actual time of departure (ATD) and the time of the 12-h DR position (see Fig. 3). By doing this, the ships's run-time position will be slowed by encountered seas. This is possible because the forecast speed

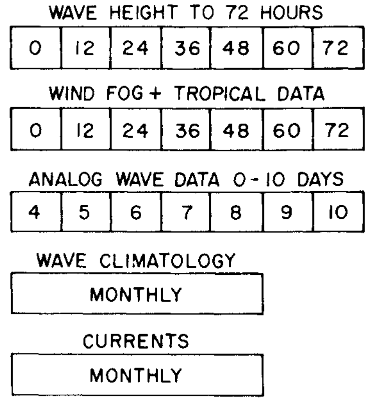

Fig. 1 Environmental arrays that are used in ship routing.

Presented as Paper 77-1613 at the AIAA/AGU/AMS/IEEE/MTS/SEG Conference on Satellite Applications to Marine Operations, New Orleans, La., Nov. 6-11, 1977; submitted Jan. 3, 1978; revision received Aug. 3, 1978.

Index categories: Computer Technology; Analytical and Numerical Methods.

*Ship Routing Programmer, Applications Software Division.

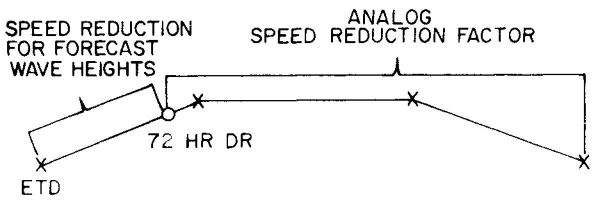

Fig. 2 Speed reduction for a ship in predeparture.

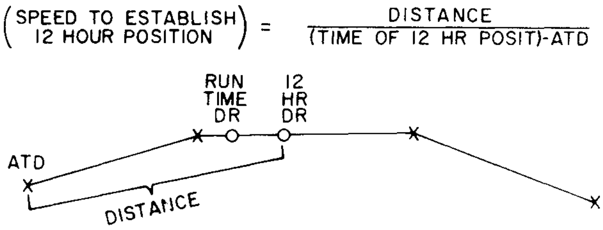

Fig. 3 Saving the speed to establish the ship's 12 hr position.

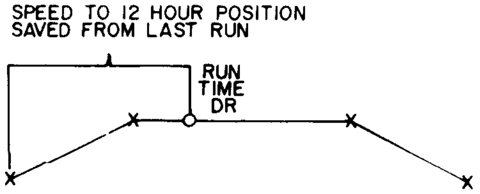

Fig. 4 Navigating the ship up to runtime.

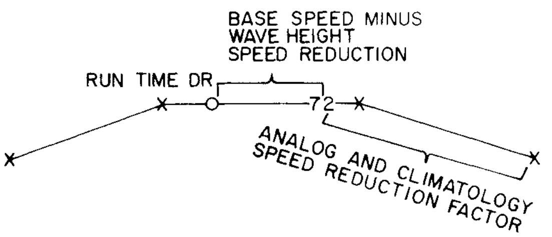

Fig. 5 Speed reduction from present position to destination.

reduction between 0 and 12 h is very accurate. Thus, it may be used to establish the run-time position 12 h in the future with good accuracy, and the ship will appear to be slowed for encountered sea heights.

After the ship has departed, the speed to establish its 12-h forecast position is saved. This speed then is used on the next run to navigate the ship up to run time. In Fig. 4, this speed has been used to establish a run-time position of the ship which is corrected for speed reduction due to encountered seas during the past 12 hr.

Once the run-time position of the ship has been established, if there are no position reports, the navigation becomes a simple matter of determining what speed to use in navigating the ship to whatever time is desired. In Fig. 5, the ship's base speed, speed profile, and wave height are used to navigate the ship up to the 72-h position, after which the analog and climatology speed reduction factors take over. The analog and climatology speed reduction factors will be used all the way to destination from the 72-h position.

The use of a base speed for a ship is fairly unrealistic, especially for a U.S. Navy ship. Therefore, a routine has been incorporated into the program to allow for speed changes of any magnitude for any duration of time. The only limitation is that there may be only 16 such changes. These speed changes may be entered into the program at any time. The routine stores the speed changes with their effective time. In navigating a ship at a specific time step, at each step the program checks to see what speed it should use. In Fig. 6,

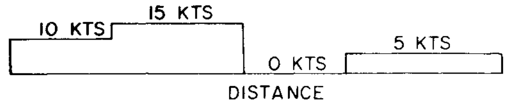

Fig. 6 Speed profile for varying ship's speed.

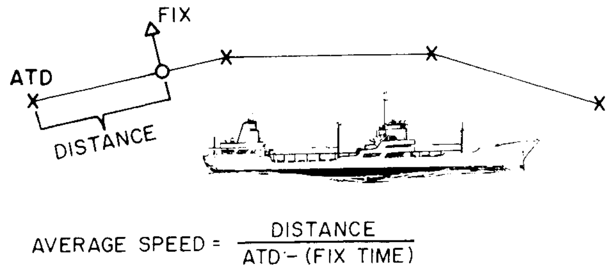

Fig. 7 Computing the average speed.

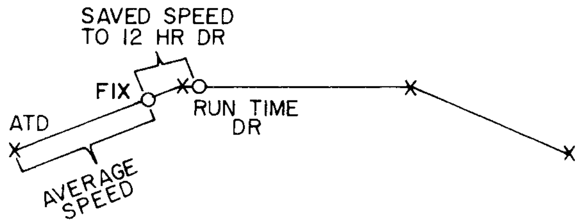

Fig. 8 Determining the run-time position.

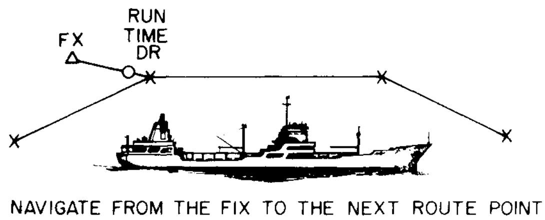

Fig. 9 Navigating from a fix.

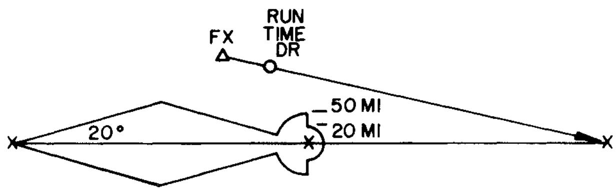

Fig. 10 Envelope used in determining the next route point.

there are several time steps for the depicted route. For each step, the program searches each speed change until it finds the appropriate speed, and this is used to navigate the ship for that time step. In the example, the speeds vary between 0 and 15 knots. The 0 speed is especially useful for U.S. Navy ships on station or a merchant ship that is dead in the water. Even though the ship's speed is 0, the ship's position still will be corrected for currents. So the program has a primitive but effective drift forecast for ships that are dead in the water.

If there is a ship position report, the navigation must be updated. This is done with a special speed called the average speed. The average speed is the speed that is required for the ship to leave the departure port at the actual time of departure and arrive at the fix position at the time of the fix. The navigation logic assumes that the base route of the actual ship is the route that the ship is on or very close to it. If the ship's route is different, the logic assumes that the base route will be modified to agree with the new route. Therefore, the ship always tries to navigate back to the base route if the fix is not on the base route. In Fig. 7, a perpendicular is drawn from the fix to the base route to get a position for an average speed computation. This speed will be used later to establish the position of the fix on the base route.

After the average speed has been obtained, it is used in determining the run-time position of the ship. This is a two-step process. First, the fix position of the ship on its track must be established, and then the ship is navigated up to run time by using the saved speed of the ship's 12-h position (Fig. 8).

This method of correcting the ship's position for speed reduction due to sea heights appears rather crude. However, it is being used operationally. It is simple, reliable, and in one study produced an average distance error of less than 5% of the total distance of the route. The accuracy is attributed to the fact that usually sea heights have little change in 12 h, and a ship usually reports daily or not at all. When a report is received, it may not be exactly on the ship's intended track. When this occurs, it is advantageous to navigate the ship from the position report instead of the run-time position along the ship's track. Figure 9 shows how it is necessary first to determine between which two route points a fix lies, so that the program can determine which route points should follow the fix position for a final ship's track to destination. With this method of navigation, it still is important to establish the run time of the ship on the new track.

If the fix is considerably off track, navigating to the next route point may not be desired. It may be far better to skip that route point and navigate to the following point. Figure 10 depicts the envelope that is utilized to make this decision. If the fix falls outside of this envelope, the navigation routine will skip the route point and route from the fix to the following route point.

In some cases, a position report may be received which is actually in the future. Such a report may be as much as 6 h past the run time of the program. In Fig. 11, the navigation routine is navigating the ship up to run time by navigating backwards for the required amount of time. Remember, the fix position is used only to determine a run-time position and a route to this destination. If the fix is off the track, the program established the track and then navigates backwards to establish the run-time position.

All of the navigation in the program is done by one subroutine that uses many different speeds. This subroutine navigates a ship along any series of latitudes and longitudes that are stored in a particular labeled common block. Through the uses of rather complicated logic, the subroutine selects the correct speed to be used for each time step and then navigates the ship along its track. This approach to the problem of navigating many different tracks with many different speeds has proven very efficient and reliable.

III. Computer-Generated Routes

A computer-generated route is a route recommendation by the computer for the entire track of the ship from present position or the departure port to the destination. Once a ship is put into the main routing program and resides in the ship's file, the computer produces a best route of its own and compares it with its previous recommendation. The original computer route remains unchanged unless the original route is flagged and the new route is not flagged or if the new route is 5% shorter. A ship is flagged when it encounters an environmental parameter that exceeds a maximum limit that is placed on it by the ship-routing duty officer. The flag is an X placed after the ship's name under the type of constraint which has been exceeded. This computer route is a global minimum time route (GMTR), which is a least-time route to the destination which does not exceed the environmental constraints of the ship. The environmental constraints that a route may be flagged for are wave height, typhoon wind radius, fog, sea height gradient, or probability of critical waves.

When the computer first was used to compute these routes, there was a problem in being able to obtain a computer-recommended route in a reasonable amount of time. A reasonable amount of time would be less than 5 min of central processor time. In the early days, times greater than 30 min

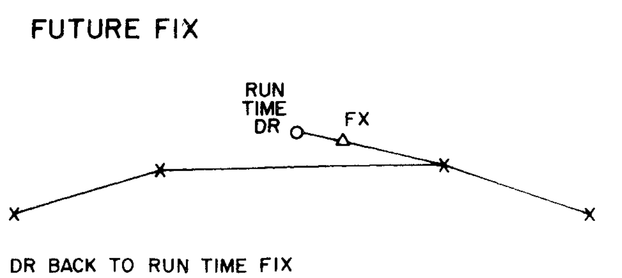

Fig. 11 Navigating back to a run-time position.

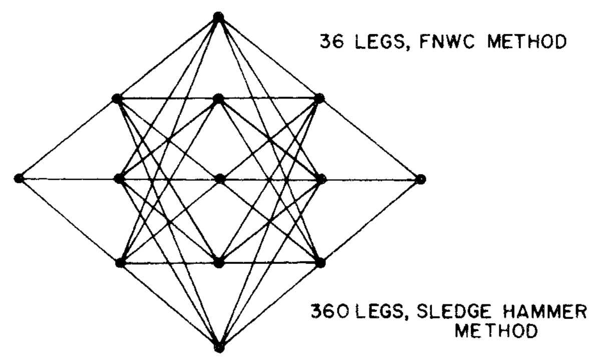

Fig. 12 Grid used in selecting a global minimum time route.

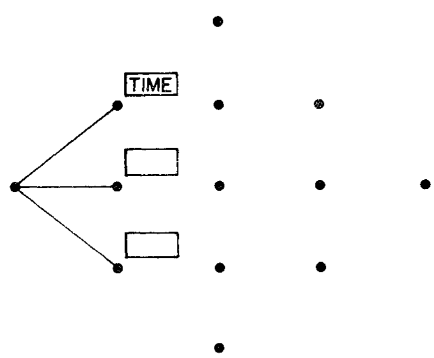

Fig. 13 Navigating to the first column in the grid.

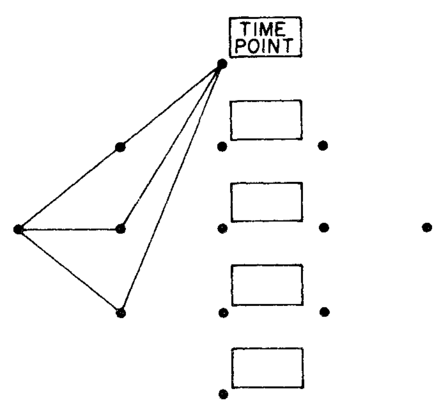

Fig. 14 Navigating to the second column in the grid.

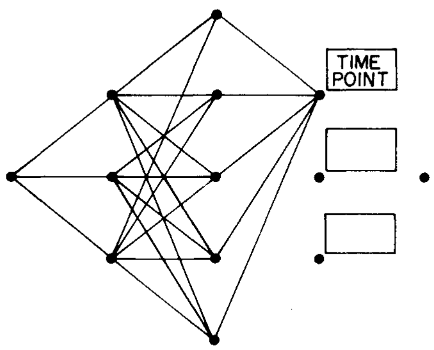

Fig. 15 Navigating to the third column in the grid.

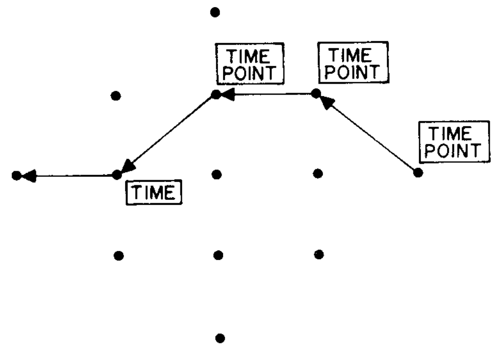

Fig. 17 Retracing the global minimum time route.

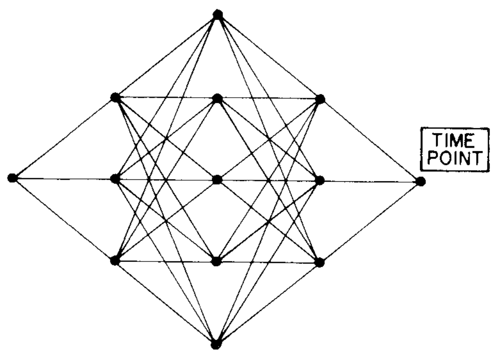

Fig. 16 Navigating to the end point in the grid.

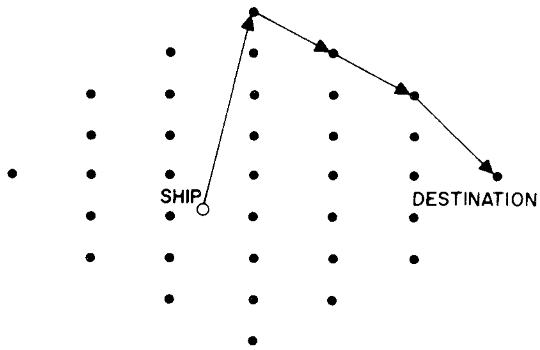

Fig. 18 Inefficient fixed grid.

were not unusual, and the author of this paper has exceeded 15 min for one route on a test run. This program has been under development for over 20 years and has undergone many changes to solve problems. The two biggest problems in computing a route for a ship are to compute a good route and to do it in a reasonable amount of time. At present, the program is averaging less than 1 min of central processor time per ship to compute both a GMTR and a phantom ship. The phantom ship is a computer-generated ship that sails in competition with the actual route. The resulting route recommendations that are produced are quite good. To compute the route, a grid such as the one in Fig. 12 is utilized to navigate the ship from point to point. Even though this is a very simple grid with a small number of points, there are a possible 90 routes through it. To navigate these routes normally would require navigating 360 legs, which would be very time-consuming. However, a method has been developed which requires navigating only 36 legs. This is the total number of legs in the grid, and each leg is navigated only once.

The first step in navigating the grid in Fig. 12 is shown in Fig. 13. The ship gets underway at the correct time, with speed reduction for forecast sea heights, and sails to the first three points in the grid. The time to get to these points then is stored.

Table 1 Effect of changing grid dimensions

Grid	Possible routes	Legs to be navigated FNWC	Sledge hammer
Original	90	36	360
Double points	3,600	132	144,000
Double columns	2,025	81	18,225
Double points, columns	129,600	360	1,166,400

After the times enroute are stored for the first column in the grid, the ship is navigated to the points in the second column in the grid (Fig. 14). At these points, the shortest of the three possible times enroute is stored with the number of the route point in the first column, through which this shortest route passes. This is done for all five points in the second column.

In Fig. 15, the same logic that was used in Fig. 14 is utilized in navigating the ship to the third and last column. Here the shortest total time enroute is stored, and the route point number in the second column through which the shortest route passes also is stored. In Fig. 16, the ship finally is navigated to the end point of the grid, and the shortest total time enroute is stored with the route point number in the third column through which the shortest or least-time route passes. It is these stored grid points that will enable us to retrace our steps and obtain the global minimum time route. The time that is stored at each point is the total minimum time enroute to arrive at that point from the start of the grid.

After the ship arrives at the end point of the grid, the number of the point in the previous column is used to retrace the path of the global minimum time route. The latitudes and longitudes of these points are extracted and stored as a complete route. An example of this is depicted in Fig. 17. The grid in this example is very simple, and the grids that are used operationally are more complicated. As the dimensions of this simple grid are increased, the number of legs which must be navigated increases very rapidly. Table 1 demonstrates how doubling both the number of columns and the number of points in a column increases the number of legs which must be navigated by the program. In our sample case, doubling these two parameters requires navigating 10 times as many legs using the Fleet Numerical Weather Central (FNWC) method. Using the sledge-hammer method would require navigating 3240 times as many legs, and this would be completely unacceptable. This is an operational program that is run twice a day, and the time to navigate 1,166,400 legs is not compatible with this type of system. Note that in this example the FNWC method of navigating the grid is fast because each leg is navigated only once. The sledge-hammer method would be

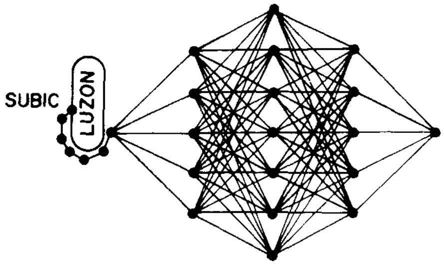

Fig. 19 Efficient variable grid with lead in leg.

to navigate each possible route one at a time, with many duplicate legs being navigated.

There are two basic types of grids which are used by this program: fixed grids and variable grids. The fixed grids are present in disk files and are for frequently used ports. These grids contain lead-in and lead-out legs to go around navigation problems such as the island of Luzon when leaving Subic Bay in the Philippines. The one big disadvantage to the fixed grid is that its points do not change as the ship moves along its track. In Fig. 18, there is an example of how this can be very inefficient.

A variable grid allows for lead-in legs and lead-out legs, and the start of the grid is the present position of the ship whether she is in port or underway. If there is a lead-in leg such as in Fig. 19, the ship will navigate the lead-in leg to the starting point of the grid, and from there the computer will determine the best route for the ship to sail to the end of the grid. Because the variable grid is recomputed each run, the resulting computer route recommendations are better than the fixed-grid recommendations. This is because the ship has a custom-made grid for each run.

The variable grid's dimensions are decreased as the ship nears its destination. This is done automatically by the computer and produces a considerable time saving. By looking at Table 1, it is easy to see that a relatively small change in grid dimensions can produce this very significant saving in time.

These grids also are used for land and ice avoidance. Before the grid is navigated to check for environmental parameters, one quick pass is made through the grid to check for land. If land is encountered on a leg, that leg is not navigated in the second pass, which checks for environmental parameters. This can increase the speed of the run considerably. For ice avoidance, it is necessary only to modify the land-sea table so that any ice is depicted as land. This is done by the ship-routing duty officer, and a chart of the modified land-sea table is produced by the computer for quality control.

The use of these grids also gives the ship router a tool to answer the question, "What would have been the optimum route for a ship?" After a ship has arrived, the 1100-word array that contains all of the historical information on the particular crossing is stacked on a magnetic tape. Later, this tape can be read into a route verification program that uses the information to determine what the optimum route would have been. This is done by making available to the program the actual seas and currents that existed during the ship's crossing, at 12-h intervals, and then navigating the ship through the grid to obtain the global minimum time route using actual sea heights, not forecast sea heights. This was done for all of the routes that were sailed with fixed grids in the year 1975, and the results still are being studied.

One of the most important features of this system is that the ship router has three knowledgeable dissenting opinions: 1) the phantom ship, which is sailing in competition with the ship's actual route; 2) the global minimum time route, which is recalculated each run for the present position of the ship; and 3) the verification route, which can be run after the ship arrives.

To compute one verification route, phantom route, or GMTR requires 45 s of central processor time. To duplicate this by hand would require navigating a ship out at 90-mile increments with a speed that was being adjusted constantly for sea heights. Do this for a few thousand routes, and select the route with the least time that avoids unacceptable weather conditions, ice, and land. You have just computed an optimum route. Without the computer, this powerful tool would not be possible.