Scientific Journals
of the Maritime University of Szczecin

Zeszyty Naukowe
Akademii Morskiej w Szczecinie

2015, 42 (114), 118–124
ISSN 1733-8670 (Printed)
ISSN 2392-0378 (Online)

Real-time winter traffic simulation tool – based on a deterministic model

Morten Lindeberg[1], Pentti Kujala[1], Jarkko Toivola[2], Helena Niemelä[2]

[1] Aalto University, School of Engineering, Department of Applied Mechanics, Marine Technology Espoo, Finland, e-mail: {morten.lindeberg; pentti.kujala}@aalto.fi
[2] Finnish Transport Agency, Winter Navigation Unit, Finland
e-mails: {Jarkko.Toivola; Helena.Niemelä}@liikennevirasto.fi

Key words: ice-breaker operations, winter navigation, simulation tool, modeling, winter, traffic

Abstract

Increasing maritime traffic, combined with the possible warming of the climate, will affect the demand for icebreaking assistance. Accurately predicting the local demand for assistance without an appropriate simulation tool is hard because of the number of variables that must be considered. This report describes a simulation tool built around a deterministic, ice-breaker movement computer model. The tool is still under development, and has not yet been tested with real data. However, preliminary test results based on self-generated input data are promising.

Introduction

Increasing maritime traffic in areas where ice-breaker (IB) assistance is needed will naturally also increase the demand for icebreaking assistance. The work load of one IB at a particular time in its assistance area is strongly dependent on ice conditions and the specific positions of the ships needing assistance. This could lead to large area- and time-specific variations in the demand for icebreaking assistance. Even under constant ice conditions, it is hard to estimate local demand for assistance solely from the estimated increase or decrease in local maritime traffic.

This report describes the working principles of a simulation tool built around a deterministic ice-breaker-movement model. The tool is programmed in MATLAB, and can be used to predict local demand for icebreaking assistance under changing ice and traffic conditions. It can also be used to predict how ship traffic will react when the local IB number is increased from one to two.

With the too in its current state of development, the fairway network (the simulation area) that is laid out, is limited to paths going from point A to point B. In this report, these paths are referred to as an IB operational area, or simply as a "line". Each line is assigned to one IB.

An example fairway network comprising three IB operational areas, each represented by a straight line, is depicted in Figure 1.

Although not visible in Figure 1, the fairway lines are equally discretized into shorter pieces in which the ice-conditions are the same at a specific time. Any kind of network can be built so long as the lines are connected to each other at their end-points. A port can be assigned to any end-point (Figure 1). One operating IB is assigned to every constructed line, and the movement of the IB is constrained by the end-points of the line.

The input of the tool consists of individual ice-data for all the discretized pieces, with the ice-data spanning the whole simulation time period. Any number of ships, at any time, can be inputted into the network through an open-ended line end-point, or "node" (see Figure 1). A ship's specific speed and its possible need for icebreaking assistance are determined from the ice conditions and the type of ship in question. The simulation starts when the first ship enters the network, and ends when the last ship leaves the network.

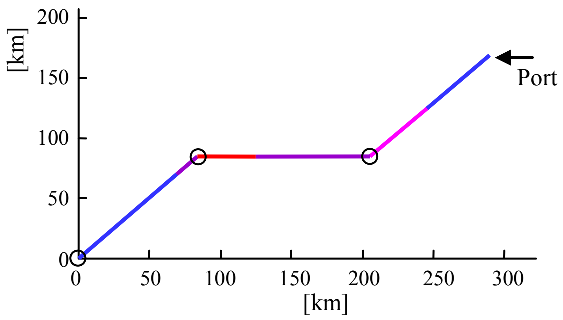

Figure 1. Visualization of three IB operational areas. The black circles are icebreakers. The ship-input-node of this network is located at the [0, 0] coordinate. A port has been assigned to the open-ended end point. The different colors represent different ice-conditions. One should remember that any kind of path going from point A to point B can be mapped onto a straight line

A previously simulation tool, designed for the same purposes as current tool, is described in (Nokelainen, Salmi & Suojanen, 2004). The Nokelainn et al, tool has some similarities with the model presented in this report, such as the ability to calculate ship speeds. However, the calculation method for ship speeds and IB assistance speeds should really be considered as separate problems and therefore addressed as model input. In addition, the Nokelainene et al. model does not treat IBs as a normal part of the ship traffic, while the model presented in this report does. Having the IBs operating in real time among ship traffic is the key feature of this model, and is the factor that most differentiates it from the model presented in Nokelainen, Salmi & Suojanen (2004).

Ice-conditions and the calculation of ship speed

Ship speed and ice-conditions

The only ship information that this model needs are the coefficients *a*, *b*, *c* and *d* of a cubic function describing the so called HV-graph. The HV-graph gives the ship speed as a function of the ice-thickness *x*: speed = $HV(a, b, c, d, x)$. The HV-graph is acquired via a ship's ice-going-capability-model, and therefore the HV-coefficients are unique for each ship. Like every other ship in the model, the icebreakers have also their own individual HV-coefficients. If HV-graphs (coefficients) of all the specific ships that should be input into the tool are not available, HV-coefficients from similar ships can be used as substitutes. Because ice thickness is the only ice-related feature that the model uses directly, ice thickness is also the only variable in the HV-graph. However, this doesn't limit the HV-graph to only considering ice thickness when determining ship speed. Different ice conditions can be translated into "ice thickness equivalents" in order to capture their effect on a ship's speed (Montewka et al., 2015). It important to note that estimating the impact on ship speed of separate ice-condition features does not make using of the tool significantly more complicated.

As the last step, the HV-coefficients should be calibrated with a simulation model run by using a ship's driving path with known travel time and ice-conditions. Nonetheless, the HV-graphs are only used by the tool as an input, and therefore they are not discussed here in more detail.

Criteria for the need of icebreaking assistance

Ship speed as expressed in the HV-graph is used to determine whether a ship needs icebreaking assistance. The threshold speed is a variable and can be chosen freely. The same value is used in this model for all ships, but an individual value for each ship is also possible.

The assistance speed is simply the speed from the icebreaker's own HV-graph. This speed is not necessarily what the speed would be in reality, as it would also depend on the ship that is being assisted. Also, if there is more than one ship being assisted, i.e. a convoy, the assistance speed will be slower. The way one defines the final assistance speed is not discussed here in more detail, as it does not really affect the workings of the simulation tool.

If local ice conditions are severe, assistance is limited to a single ship at a time; that is to say, no convoys can be formed when moving through a severe ice pack. The icebreakers of the tool will take this into account while scheduling the optimal assistance mission. The criterion for severe ice conditions is a limit-speed value from the icebreaker's HV-graph.

The constructed model for the icebreaker movements

Main principles of the icebreaker movements

The movements of icebreakers are based on the waiting times of the ships: a ship gains waiting time when it cannot move in the ice, and it is not being assisted. A ship gains waiting time at the same pace as the real-time flows. If one would like to prioritize specific ships, it is of course possible to assign a higher weight value to such ships.

The icebreaker chooses the assistance mission that minimizes the cumulative ship waiting times in the time period spanning the duration of the chosen assistance mission. The icebreaker considers only the waiting times of ships that are within the endpoints of the specific straight, that is to say, within the operational area of the icebreaker.

Single-way (SW) and double-way (DW) missions

The icebreaker makes an SW mission when it only has information of waiting / or incoming ships into one of the end-nodes of the line. If the icebreaker leaves one or more ships behind (as can happen when ships do not arrive before the IB begins assisting an earlier ship) when choosing its mission, then the duration of assistance will be increased to include time needed to return for the later ships, such that the later-arriving ships gain waiting time until the IB arrives. This is demonstrated in Figure 2.

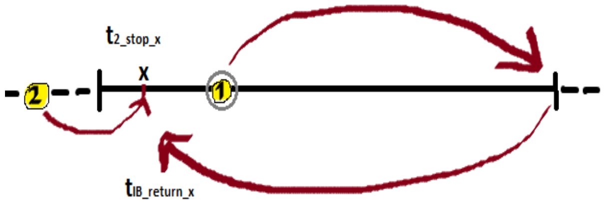

Figure 2. SW-mission. Yellow dots are ships and the gray circle is the IB. The IB operational area is the marked black line. The stop position of ship 2 is "x" and stop time is "*t2_stop_x*". IB returns to "x" at time "*tIB_return_x*"

Figure 2 shows an example of a SW-mission where the IB chose only ship 1, while having the information of ship 2 entering the operational area and stopping at position "x" at time *t2_stop_x*. The waiting time gained by ship 2 in the example mission would have been *tIB_return_x* – *t2_stop_x* (assuming IB arriving later than ship 2 stops).

The IB makes a DW mission when it has information of ships in both node A and node B. This means that it will choose one or more ships from A-node, and similarly, one or more ships from B-node. The node-order and the number of chosen ships from each node (single assistance or convoy) that minimizes the cumulative waiting time, will be the chosen mission. The possible node-orders are A→B→A or B→A→B. So, for example, if the IB has information of 3 ships in each of its nodes, there will be 18 $((3 \times 3) \times 2)$ possible missions to choose from. Figure 3 demonstrates a DW mission.

Figure 3 shows an example of an A→B→A DW-mission. The example situation is the same as in Figure 2, except that now the IB also has information of a ship arriving at its B-node. The cumulative waiting time of this example mission is given by (*tIB_arrive_B* – *t3_stop_B*) + (*tIB_assist_end* – *t2_stop_x*). It should be noted that *tIB_arrive_B* is chosen here for convenience to be equal to the assist start-time from B-node. The important thing to notice is that the stop position of the IB's second assist part is not considered in the waiting time calculation for ship 2; it is just a coincidence that it looks like the positon would be "x" in the picture. Note: in the example it is assumed that ship 1 did not gain any waiting time, but if it would have gained, it would be calculated in the same way as for the other ships.

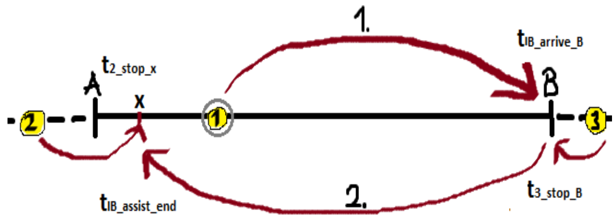

Figure 3. DW-mission. Yellow dots are ships and the gray circle is the IB. The IB operational area is the marked black line. The stop position of ship 2 is "x" and stop time is "*t2_stop_x*". The stop position of ship 3 is the B-node and the stop time is "*t3_stop_B*". The first assist end-time is "*tIB_arrive_B*" and the second assist end-time is "*tIB_assist_end*"

In a DW mission, the ships that are not assisted (e.g., ship 2 of Figure 3) will gain waiting time until the IB finishes its mission (*tIB_assist_end* of Figure 3). This differs from the SW mission where the waiting-time-gain-period stops when the IB returns to the later arriving ship (Figure 2). The reason for the different method is that, for example, if the IB left some ship unassisted (due to its late arrival) in the B-node of Figure 3, then the IB's return time to this imaginary ship is not known with certainty at the starting time of the first mission. This is due to two reasons: (1) the next mission could be a DW mission with two or more ships from the side of the initial starting node, and this would require the calculation of the next mission at the time of calculation of the current mission; or (2), the mission duration of a DW mission is long enough so that the likelihood of new incoming ship information into the starting-node is high, such that it would not even be possible to calculate the IB's return time with certainty at the time in question. Of course, the IB return time calculated in a SW-mission is not completely certain either (Figure 2), but its likelihood of being correct is much higher because the mission duration is shorter. Nonetheless, the method used for the waiting time calculation seems to work well.

The main reason for the manner of construction of the DW-mission is because the IB will process just enough information to be able to choose the mission that seems most natural in the real word. Lengthening the decision-chain (when possible) to triple-way (A→B→A→B) would probably not yield any benefits.

Communication between the icebreakers

The communication between the fairway-lines (IB operational areas) is of crucial importance for the simulation model to work properly. The model will immediately start to differ from reality if the IB-communication is not smooth.

The IB receives information of ship arrival times at end-points of the lines (nodes A and B). This information is delivered to the IB before the ships actually enter the IBs operational area. The arrival-time information is delivered by neighboring IBs. The time when the information is passed on to the neighbor(s) I,s at latest, the time when the IB starts to move (IB_start_time), and this is not necessarily the start time of the scheduled assistance.

The time difference between IB_start_time and the time the assisted ships arrive to the neighboring node (arrival_time), is not always enough for the neighboring IB to react in time to ensure optimal IB movements. Therefore, an adjustable parameter is used in the model that lengthens the time that the information receiving IB has to react to the information it receives. The parameter can take real-time values ranging from 0 hours up to about 5 hours before the scheduling scheme of the program gets distorted and the simulation run crashes. The limiting value depends on the traffic and the fairway network. However, the "advance-time" that can be achieved within the limits of the parameter value seems to be enough. The advance-time that the neighboring IB gets will be given by: *advance-time = "parameter value" + arrival_time – IB_start_time*. It should be noted that *arrival_time – IB_start_time >= duration of assistance*.

When using the communication method just described, there is no need for the neighboring IBs to be aware of each other's positions, because the only thing that really matters is when the ships needing assistance enter the IB's operational area.

It should be noted that only neighboring IBs communicate with each other. Communication reaching further than neighboring IBs' operational areas would significantly complicate the structure of the program, and the deeper communication would only be beneficial if the *advance-time* this communication made possible would not be enough.

Stop positions of ships that need assistance

Ice-conditions can of course be such that the ship can manage on its own beyond the end-point of the IB's operational area. The ship will stop naturally at the position where its speed drops below the limit-value.

If another ship arrives to the position where another ship is waiting for assistance, the new ship will not pass the earlier ship even if its ice-going capability would allow it to do so. This is because the IB would not assist the stronger ship first in the real world either. In this situation, the IB of the tool will, if possible, form a convoy, and the assistance start position will be the position of the earlier ship. If a convoy is not formed and the IB assists the earlier ship first, then of course the stronger ship will continue as far as it can manage on its own.

The IB-algorithm

The part of the simulation program that plans the movements of the IB, and that also shapes the IB's operational area (the straight line), is referred to as the IB-algorithm. A flow diagram of the operation of the IB-algorithm is shown in Figure 4.

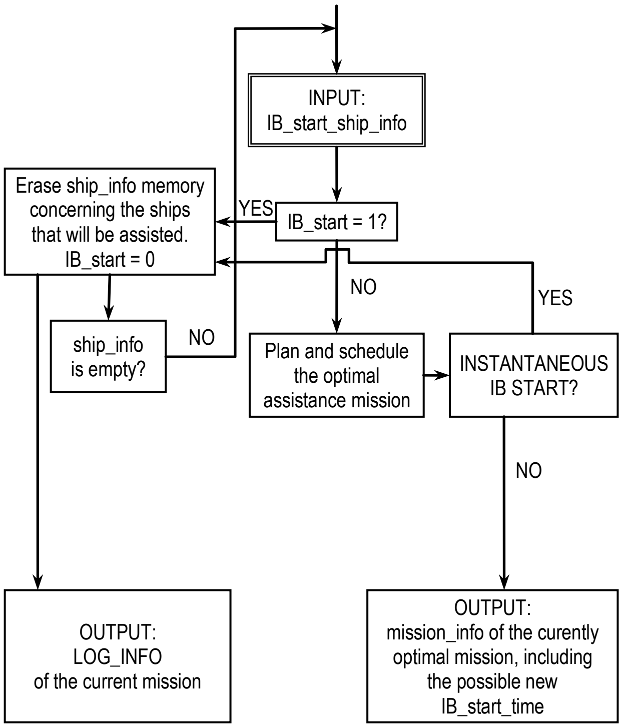

Figure 4. Flow diagram of the IB-algorithm

The IB-algorithm is the building block of the main program of the simulation tool. The fundamental function of the main program is to schedule the *IB_start_times*, outputted by the IB-algorithms, and to put the outputted ship-log information into memory.

As can be seen from the flow diagram, the structure of the IB-algorithm is fairly simple. The main idea is that the algorithm is used when the variable *IB_start* is zero. The algorithm then computes the optimal way to assist the new ship(s). If the variable value is one, then the IB will start a scheduled assistance mission, which has been planned by the algorithm in the past.

The box in the middle, on the left-hand side of the flow diagram, checks whether the IB chose to assist all of the ships that it had information for. If not, then the algorithm will be run, with the input consisting of the ships that it didn't choose to assist.

The box on the top of the right hand side of the diagram determined whether the IB must start instantly. This action occurs when the optimal assistance mission (calculated by the algorithm) would be to assist the new ship(s) (the input) first, and the IB will not be able to sail to the new ship's stop position in time. If the value of the adjustable parameter (sub-topic C) is, for example, set to zero, then the instantaneous IB start would occur frequently, and the movements of the IBs would not be optimal.

Further development

This report has so far only covered the parts of the tool that have been implemented. However, a fairway building-block for the tool that covers the area surrounding a fairway junction (i.e. three fairway paths with variable lengths going out from one point) is under development. This building block can be assigned to one or two ice-breakers. Figure 5 demonstrates a junction building-block with one IB.

The IB-missions in a junction with one assigned IB (Figure 5) are calculated with the same principles already presented for SW vs. DW missions. The difference is that now there will is the possibility of having ship-information in all three nodes. This will make the decision process a bit more complicated, and there are probably many viable ways to define effective decision rules. No matter how the decision rules are defined, the waiting position (Figure 5) in the middle of the junction will be a feature of the junction-building block: if the time separation (not marked in Figure 5) between the ships is small enough, then the IB will naturally take both ships in a convoy and leave one of the ships waiting at the waiting position (if the ship could not manage on its own to its destination) while assisting the other ship to the point where it can manage on its own.

The junction building-block with two assigned IBs will also be developed. This is also naturally the most challenging building block to design, and this report will not go into the details of the possible principles of defining the IB movement rules. Figure 6 demonstrates one possible operation of the IBs in the junction block.

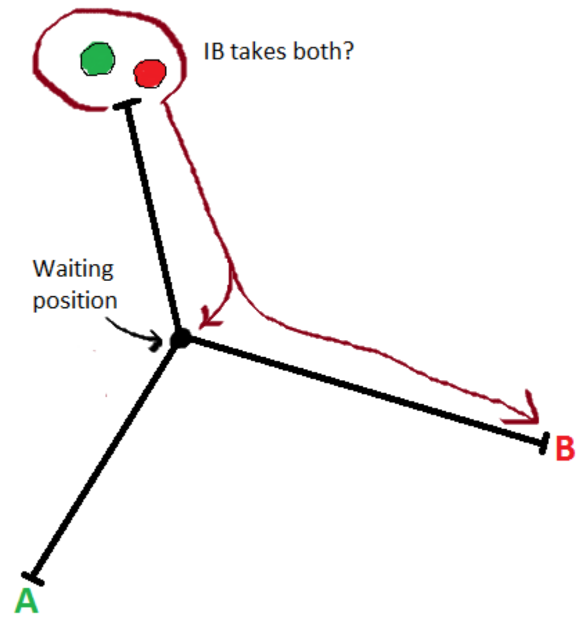

Figure 5. Junction building-block with one ice-breaker (IB not visible). The green dot is a ship with the A-node as destination. The red dot is a ship with the B-node as destination

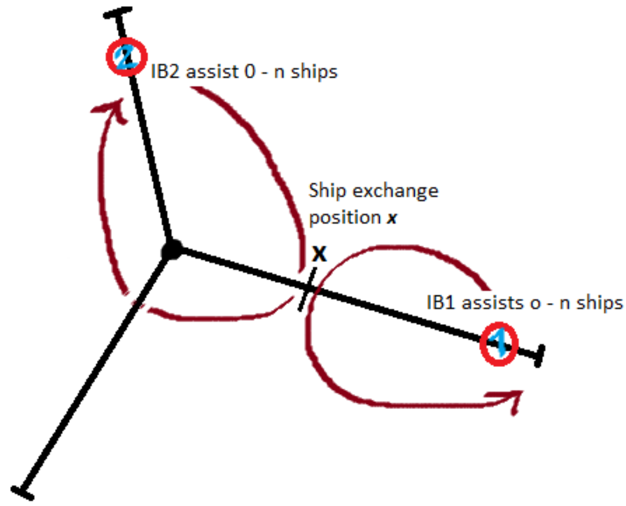

Figure 6. Junction building-block with two ice-breakers (red circles). The figure is a representation of one possible mission where the IBs cooperate to minimize the ship waiting times

The situation presented in the junction block of Figure 6 demonstrates a mission in which each IB assists 0-n ships from the node area closest to their position and then exchanges ships at position "x" (optimal position of course). This is of course only one of the possible missions, but it is special because it will probably be possible, with little manipulation of the program code, to "erase" one of the junction legs. This would transform the junction block to a "new" building-block: a straight line with two assigned IBs. Of course, all of this requires that the junction with two IBs is constructed successfully first.

Results

The simulation model has not yet been tested against real-life data, so it is hard to judge how realistic the constructed model is. However, the model has been tested with self-generated ice-data along with a random ship time-schedule. As far as the model is concerned, it is working exactly as it was designed, so in that sense, the results are positive.

The figures below show some shots of the visualization of a particular simulation run.

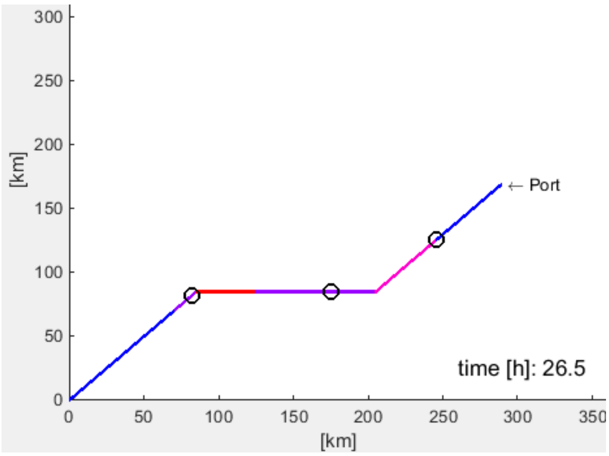

Figure 7. Simulation at time 26.5 hours. Circles are IBs

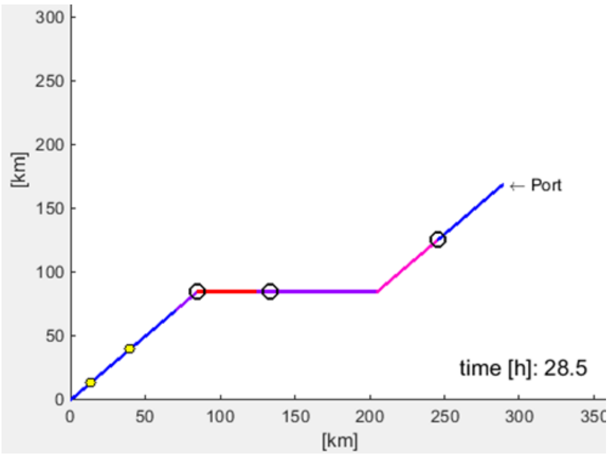

Figure 8. Simulation at time 28.5 hours. Circles are IBs. Yellow dots are ships

In Figure 7 the IB of the middle IB-line (horizontal in the visualization) is moving towards its left node. Looking at Figure 8, we can see that the middle IB was moving because it had received ship-information from its left neighbor IB. The IB started moving even before the ships entered the operational area of the IB that sent the ship-information. This demonstrates the effective communication between the IBs.

The above example is important because the ice conditions and the layout of the fairway lines of the example simulation run make up a situation where the model will be most likely to fail as a model: ships need assistance only on the last 1/6 of the line and the serving IB was really close to the stop point of the ships. At the same time, the neighbor IB is at the farthest possible location from its next assistance starting position. This setup will cause the middle IB to receive the ship-information really late, and with a bad communication scheme the IB will not start moving when it actually should. However, in this case, the advance-time parameter could be set large enough to accommodate the situation. If the length of the middle fairway line and/or its ice conditions were more severe, then the parameter value used would not have been large enough, and as the value the actually was used was close to its maximum possible value (which is case specific), it would be impossible to get the middle IB to start in time with the current version of the program.

Figure 9 demonstrates a case were the IB makes a DW-decision.

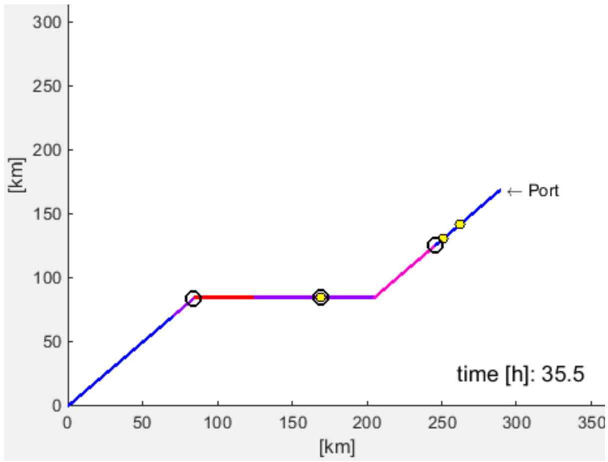

Figure 9. DW-mission. Circles are IBs. Yellow dots are ships

In the situation displayed in Figure 9, the middle IB is assisting a ship to its right side while the left-most IB is staying still in the middle of its line. At the same time two ships are moving towards the IB where they will reach their stop point. Now, the left most IB could have made it in time to its left node where the middle IB will end its assistance before the two other ships will reach their stop point. This simply means that the left IB calculated that it will minimize the ship waiting times by first waiting for the two ships.

Conclusions

This simulation model has not been compared to other models as there are not any highly comparable models available. And said before, this model has not yet been tested with real-life data. However, the tool seems to work well within its own domain. The calculation procedures for ship and IB speeds, as well as for ship stop points, is a separate model from this simulation model. This means that when judging this simulation model, one should try to consider these two model functions separately.

The main focus of this report is on the working principles of the simulation tool. The tool's most important feature is the fairway network that it lays out piece by piece. This will introduce the discontinuity points in the network. These points are probably the most likely part of the model that might cause un-realistic IB movements. Effective communication between the IBs is the one and only thing that will "blur out" the discontinuity points.

The un-realistic feature caused by points of discontinuity cannot be completely eliminated so long as the fairway network is constructed of shorter pieces. However, the junction building-block with two IBs could possibly be used in a way so that it would connect two IB-lines together without a discontinuity point. Depending on the amount of pieces used in the fairway network, this would reduce the number of discontinuity by about half.

Acknowledgments

Tom Mattsson from Aker Arctic Technology Inc. provided the HV-graphs. The Swedish Maritime Administration is currently funding this part of the project (WINMOS).

References

1. Montewka, J., Goerlandt, F., Kujala, P. & Lensu, M. (2015) Towards probabilistic models for the prediction of a ship performance in dynamic ice. *Cold Regions Science and Technology*. 112. pp. 14–28.
2. Nokelainen, A., Salmi, P. & Suojanen, R.A. (2004) *Suomen talvimerenkulun kehittäminen, Jäänmurtajatarpeen simulointityökalu*. Helsinki: Merenkulkulaitos.