Master’s degree thesis

LOG950

Fleet Sizing for Pipe Lay Support Vessels

Lidziya Kukharava Illia Zabrodski

Number of pages including this page: 59

Molde, 24.05.2019

Molde University College
Specialized University in Logistics

Mandatory statement

Please complete the mandatory statement by placing a mark ***in each box*** *for statements 1-6 below.*		
1.	I/we hereby declare that my/our paper/assignment is my/our own work, and that I/we have not used other sources or received other help than mentioned in the paper/assignment.	☑
2.	I/we hereby declare that this paper 1. Has not been used in any other exam at another department/university/university college 2. Is not referring to the work of others without acknowledgement 3. Is not referring to my/our previous work without acknowledgement 4. Has acknowledged all sources of literature in the text and in the list of references 5. Is not a copy, duplicate or transcript of other work	Mark each box: 1. ☑ 2. ☑ 3. ☑ 4. ☑ 5. ☑
3.	I am/we are aware that any breach of the above will be considered as cheating, and may result in annulment of the examination and exclusion from all universities and university colleges in Norway for up to one year, according to the Act relating to Norwegian Universities and University Colleges, section 4-7 and 4-8 and Examination regulations section 14 and 15.	☑
4.	I am/we are aware that all papers/assignments may be checked for plagiarism by a software assisted plagiarism check	☑
5.	I am/we are aware that Molde University College will handle all cases of suspected cheating according to prevailing guidelines.	☑
6.	I/we are aware of the University College's rules and regulation for using sources	☑

Personal protection

Personal Data Act

Research projects that processes personal data according to Personal Data Act, should be notified to Data Protection Services (NSD) for consideration.

Have the research project been considered by NSD? ☐ yes ☑ no

- If yes:

Reference number:

- If no:

I/we hereby declare that the thesis does not contain personal data according to Personal Data Act.: ☑

Act on Medical and Health Research

If the research project is effected by the regulations decided in Act on Medical and Health Research (the Health Research Act), it must be approved in advance by the Regional Committee for Medical and Health Research Ethic (REK) in your region.

Has the research project been considered by REK? ☐ yes ☑ no

- If yes:

Reference number:

Publication agreement

Agreement on electronic publication of master thesis

Author(s) have copyright to the thesis, including the exclusive right to publish the document (The Copyright Act §2).

All theses fulfilling the requirements will be registered and published in Brage HiM, with the approval of the author(s).

Theses with a confidentiality agreement will not be published.

I/we hereby give Molde University College the right to, free of charge, make the thesis available for electronic publication: ☐ yes ☑ no

Is there an agreement of confidentiality? ☑ yes ☐ no

(A supplementary confidentiality agreement must be filled in)

- If yes:

Can the thesis be online published when the period of confidentiality is expired? ☑ yes ☐ no

Date:

Høgskolen i Molde

STANDARD AGREEMENT

This agreement is between

Lidziya Kukharava, Illia Zabrodski…………………………………………….. (Student(s)),

Irina Gribkovskaia…………………… (Faculty Advisor at Molde University College),

Pontifical Catholic University of Rio de Janeiro……………… (Company/Institution),

And Molde University College/MSc Logistics Program Coordinator, concerning the use of specifications and results reported in the Master's degree thesis in accordance with the study plan for the Master's degree program in Logistics at Molde University College.

1. The student will complete the work assigned for the Master's degree thesis in cooperation with the company/institution
(Optional):……………………………………………………………………………………………

The title of the thesis is: Fleet Sizing for Pipe Lay Support Vessels.

2. The student has copyrights to the thesis. Those copies of the thesis submitted for evaluation along with descriptions and models, such as computer software that is included as part of or as an attachment to the thesis, belongs to Molde University College. The thesis and its attachments can be used by the College for teaching and research purposes without charge. The thesis and its attachments must not be used for other purposes.

3. The student has the right to publish the thesis, or parts of it, as an independent study or as part of a larger work, or in popularized form in any publication.

4. The company/institution has the right to receive a copy of the thesis with attachments, and the College's evaluation of it. The company/institution will have three (3) months from the time the thesis is submitted to the College for censoring to determine whether a patent is possible and to apply for a patent for all or part of the results in the thesis. The specifications and results in the thesis can be used by the company/institution in its own activities.

5. An additional confidentiality agreement may be entered into between the parties, if the company/institution sees this as necessary.

6. Each part of the agreement should have one copy of the agreement. The final signature should be from the Program Coordinator/Dean validating the agreement.

Place……Molde……… Date of final signature……23.05.2019……

Student(s) — Faculty Advisor: Irina Gribkovskaia

Company/Institution: Silvio Hamacher, Gerente de Área, Tecgraf / PUC-Rio — Program Coordinator / Dean: Berit Irene Helgheim

Høgskolen i Molde

SUPPLEMENTARY AGREEMENT/CONFIDENTIALITY AGREEMENT

(Supplement to Point 5 of the Standard Agreement)

This agreement is between

Lidziya Kukharava, Illia Zabrodski... (Student(s)),

Irina Gribkovskaia.. (Faculty Advisor at Molde University College),

Pontifical Catholic University of Rio de Janeiro....................................... (Company/Institution),

and Molde University College/MSc logistics Program Coordinator. This is a supplement to the Standard Agreement, which defines the use of specifications and results reported in a Master's degree thesis in accordance with the study plan for the Master's degree program in Logistics at Molde University College.

1. As stated in Point 5 of the Standard Agreement the parties named above agree to limit access to the Master's degree thesis for 5 Year(s) from the date of this agreement. The limit should be 1 to 5 years, for longer confidentiality periods, contact the program coordinator for a special agreement.

2. The reason for keeping the thesis results confidential is so that the company/institution can avoid giving away information that would give others a competitive advantage.

3. The thesis must be submitted to Molde University College. It will be kept in a secure room in the main archive at the College during the period agreed upon. The thesis can be borrowed during this period if a written request is submitted and the company/institution gives permission. Academic staff members can access the thesis after signing a declaration of secrecy. After the period of confidentiality is ended the thesis will be placed in the library for general use.

4. Four copies of this agreement must be signed so that each party will have a copy. The agreement is valid when it has been approved and signed by the program coordinator or Dean at Molde University College.

5. The confidentiality agreement does not exclude members of the academic staff from being present at the final oral presentation of the thesis.

Place....Molde.................. Date of last signature.......23.05.2019...........

Student(s)

Irina Gribkovskaia

Faculty Advisor

Company/Institution

Silvio Hamacher
Gerente de Área
Tecgraf / PUC-Rio

Agreement approved by:

Berit Irene Helgheim

Program Coordinator/Dean

Preface

This master thesis was written at Molde University College - Specialized University in Logistics as a final step in achieving Master of Science (MSc) in Logistics degree in Logistics Analytics specialization.

The topic of the thesis is "Fleet Sizing for Pipe Lay Support Vessels" which was formulated with the help of Pontifical Catholic University of Rio de Janeiro. We want to take the opportunity to thank our colleagues from this university, Victor Cunha and Rafael Martinelli, for providing us with details about the existing problem and data used in computation experiments.

We want to thank our supervisor Irina Gribkovskaia who not only advised us to use the unique opportunity to study at Molde University College but also inspired us by her great professionalism and skills on our long and very interesting learning path.

We want to express our gratitude to Yauhen Maisiuk, PhD student at Molde University College, for valuable advice in the simulation model development.

Finally, we want to express our huge appreciation to Molde University College for being admitted to the Master program in Logistics and invited to study in Molde.

Summary

The research area of this thesis is the upstream part of offshore oil and gas logistics. Offshore oil production requires specialized support activities, which may be achieved with relevant support vessels. An important phase for oil field development is the launching of production lines and its connection to subsea platforms. These activities are performed by pipe lay support vessels which represent expensive resources needed by petroleum companies. Determination of the size of such fleet requires proper planning because these vessels can be hired on both long- and short-term contracts which differ by price. Delays of activities implemented by vessels lead to financial impacts that can reach one million dollars per day. That is why companies sometimes use short-term contracts, which in turn are expensive resources. Hiring an optimal number of vessels on long-term contract is beneficial for companies as it prevents them from financial impacts caused by operational disruptions and more expensive short-term contracts. This fact revealed a new problem in logistics which we called Fleet Sizing for Pipe Lay Support Vessels.

This master thesis aims to develop a methodology solving Fleet Sizing for Pipe Lay Support Vessels with uncertain operations durations and a decision support tool implementing this methodology.

The proposed methodology involves a combination of optimization and simulation methodologies. It suggests schedules of voyages with a minimized number of vessels taking into account the stochastic character of operations implemented by them. The developed tool iteratively builds schedules and checks them on robustness step by step decreasing the fleet size.

Contents

1 INTRODUCTION

Energy plays a huge role in people's lives. Roughly one-third of the world's primary energy comes from such primary fuel as oil. Renewable energy technologies still overcome numerous barriers to become competitive with oil. These barriers include higher capital costs, siting and transmission. But the most formidable barrier relates to the well-established and well-understood nature of existing technologies for oil production what makes oil one of the major market players. *Fleet Sizing of Pipe Lay Support Vessels (PLSV) with uncertain operations durations* is a *real* problem for the upstream part of offshore oil and gas logistics. Offshore logistics is considered to be more complicated than onshore due to the characteristics of offshore operative which presents lots of logistic challenges, such as marine environment, limiting infrastructures and space restrictions, hard climatic conditions which lead to operational disruptions. If the increase in the number of platforms and their physical distance are taken into account, then this complexity increases considerably. Flexible solutions to the problem mentioned above will help oil companies to overcome financial impacts caused by operational disruptions. No work in the literature has simultaneously considered a fleet sizing problem with stochastic processing times and schedule determination.

The research problem is *relevan*t. Oil is the dominant fuel for transportation, and this tendency is expected to continue. The problem is relevant to the oil industry for several reasons. Thus, reducing the cost of a fleet of vessels to hire is an important long-term factor determining a company's budget. Under real-world conditions (weather, bureaucratic procedures) activities performed by PLSVs often have inconstant duration time. As a result, there can happen a situation when not all planned wells are processed in time and companies have to hire PLSVs on a short-term contract which is much more expensive (a few hundred thousand dollars each vessel per day) in comparison to the long-term fleet. It can have a damaging impact on the company's economy. That is why companies are interested in searching for feasible solutions of such problem giving answers on how many vessels to hire on long-term contract taking possible changes in operations processing times into consideration.

The problem under consideration combines two tasks into one: scheduling and fleet sizing under conditions of uncertainty which evaluate from stochastic factors influence.

Scheduling resources such as crew, vessels or vehicles is a *challenging* task because such problems have a combinatorial structure, and several operational constraints must be observed. In general, the PLSV Scheduling problem is NP-hard so metaheuristics can be observed. The research problem in this thesis integrates such stochastic factor as vessels processing time. Thus the fleet-sizing problem under consideration requires stochastic approaches. One of such approaches implies using simulation modelling. The difficulty of it lies in the fact that activities are performed by vessels in parallel. Therefore, the implementation of parallel processes with access to common resources is required.

1.1 Research objectives

The *goal* of the research is to develop a methodology offering optimal fleet sizing on long-term contract considering stochastic operations durations and giving a schedule of PLSV activities, and the decision support tool implementing this methodology. This tool should also be able to determine a possible total delay of operations which can occur for the specified number of vessels in the schedule so as to give companies an ability to analyze the financial value of risks caused by operations delays and decide whether the chosen fleet size is suitable enough. The planning horizon here can be taken any as the decision support tool should be able to propose solutions for any input time range.

1.2 Thesis organization

In Chapter 2, a real-life *Fleet Sizing of Pipe Lay Support Vessels with uncertain operations durations* problem is described, the idea of dividing it into subtasks is discussed.

In Chapter 3, the literature about the most relevant problems about scheduling is discussed. It addresses scheduling on parallel machines in general case, dispatching rules for solving this problem and rescheduling of a pipe-laying support vessel fleet in charge of subsea oil well connections under deterministic conditions. In this chapter, we observe the similarity of this problem with the research problem and difference from it. Here we also discuss the discrete-event simulation methodology, how it was applied in other offshore logistics problems, which logic models were used by different authors so to solve their specific problems.

In Chapter 4, our new methodology is explained. The development of the decision support tool and the structure of input data for it are described. Here we fully explain the two phases of the tool development responsible for optimization and simulation.

In Chapter 5, the computational experiments and the output results of the proposed decision support tool in a practical case are presented.

Finally, in Chapter 6, we present the main conclusions of the study and ideas for future research.

2 FLEET SIZING FOR PIPE LAY SUPPORT VESSELS WITH UNCERTAIN OPERATIONS DURATIONS

Offshore oil production requires specialized support activities, which may be achieved with relevant support vessels. Oil is drained through pipelines connected to subsea wells (Figure 2.1).

Figure 2.1 Pipelines connecting subsea wells and platforms

An important phase for the development of an oil field is the launching of risers (production lines), and its connection to subsea platforms. The vessels responsible for the launching of risers and its connection to subsea platforms are the PLSVs (Figure 2.2). These resources are essential to be managed. Such vessels connect flexible lines and risers to wells and platforms. But this process is divided by several activities. At the beginning of the operation the vessel needs to load the lines and the required equipment at the port. Then the vessel moves to an assigned place in order to initiate the launching phase. Delays at the beginning of well production lead to financial impacts that can reach one million dollars per day (Cunha, 2018). That is why the connection of pipelines performed by PLSVs is a very important process in oil exploration and production.

Figure 2.2 Pipe lay support vessel

2.1 Problem description

The current problem considers a determined amount of wells to be connected to platforms. Each well has a due date to be processed. There are several activities performed by the PLSVs for the company's needs. Each of the vessels can perform more than one activity per voyage when it goes from the port and back. The objective is to minimize the financial impacts caused by operational disruptions, i.e. minimize delays of wells production. All operational restrictions such as availability of vessels, vessels capacity, due dates of wells, release dates of operations can be treated as mathematical constraints. The Fleet Sizing problem must give an answer on how many vessels the company should hire per long-term contract on minimum to meet a given production plan. The problem under consideration contains stochastic factors influence. These factors relate to bureaucratic procedures and climatic conditions such as waves height which in turn make operations duration times undetermined and therefore stochastic. So as to solve the Fleet Sizing Problem under conditions of uncertain operations durations, it's needed to execute a subtask - to build a deterministic schedule of PLSVs. It means to solve the PLSV Scheduling Problem. The PLSV scheduling problem can be modeled as an identical parallel machine scheduling problem, which consists of *I* jobs ($1 \leq i \leq I$) to be processed on *K* machines ($2 \leq k \leq K$), so that each job is performed only once by one of the available machines (Cunha, 2018). But as the number of machines in the problem is undefined, number *K* is assumed to be no less than the number of operations *I*. So the observable subtask consists of vessels and activities for connection operations which are equivalents of machines and jobs respectively. Each job is associated with one or several production wells, therefore for each well there is a subset of jobs associated with it. The completion time of each well is given by the completion time of the last operation associated with it. Ideally, a well should be processed until its due date d_j. To execute an operation, a vessel must be loaded at the port with the material required for the operation, such as pipelines and other equipment, navigate through all well locations associated with that operation, process them and navigate back to the port. This cycle is called a PLSV voyage, which can comprise more than one operation. However, as each operation has a percentage occupation rate l_i of the ship deck, the total number of operations in a voyage must respect the vessel cargo capacity. The due date of the operation will be considered to be equal to the earliest due date of wells associated with that operation.

In the current problem, each operation has a processing time p_i, which is independent of its position in the voyage or of the vessel that executes it. Besides, an operation has a release date r_i, related to the availability of the material needed for its execution.

The vessel loading time at the port and the navigation time to the wells are treated as setup time, and its duration depends on the operation type f_i that has its own setup duration s_f. In the exploratory regions, where the wells are geographically close to each other, the navigation time between them is very similar, taking only a few hours for a PLSV to travel from one well to another. The execution time of pipe-laying activities in turn is counted in days. For that reason, we will include the navigation time between the wells in the processing time of the job related to them. The only navigation considered separately is between the port and the exploratory region, for which we will apply a deterministic duration and which is included into setup duration.

In order to present the problem notations and formulate essential details, let introduce a mathematical model developed for the PLSV Scheduling Problem with the respective definitions of sets, subsets, parameters and decision variables. It is worth noting that the model here uses a family scheduling formulation considering an order index to define the sequence of tasks for a vessel. The families of the vessels and some other parameters are considered here.

Sets:

$j \in J$ set of wells

$i \in I$ set of operations

$k \in K$ set of vessels

$o \in O$ set of orders

$f \in F$ set of families

Subsets:

K_i a subset of vessels eligible to execute operation i

F_i a subset of families associated with operation i

J_i a subset of wells associated with operation i

P_i subset of operations i' that must precede i

Parameters:

p_i operation processing time

l_i operation load rate

r_i operation release date

d_i operation due date

B_k vessel lower window

F_k vessel upper window

w_j daily well production

w_i relative daily production of operation i, equals to $\max_{j \in J_i}\{w_j\}$

s_f setup duration for family f

L_k^{max} the maximum load capacity of vessel k

R^{max} maximum release date: $\max_{j \in N}\{r_j\}$

M a large number

α weight of machine tardiness in the objective function

β weight of operations tardiness in the objective function

Variables:

X_{ifk}^{o} equals 1 if operation i, from family f, is scheduled in the o-th order on vessel k, 0 otherwise

Y_{fk}^{o} equals 1 if a setup from family f, is scheduled in order o, on vessel k, 0 otherwise

S_{k}^{o} starting time of o-th order on vessel k

S_{i} starting time of operation i

L_{k}^{o} Total load of o-th order on vessel k

R_{k}^{o} release of o-th order on vessel k

C_{i} completion time of operation i

C_{j} completion time of well j

T_{k} Tardiness on vessel k

T_{j} Tardiness of job j

Objective function:

$minimize\colon \sum_{j \in J} w_j T_j + \sum_{k \in K} T_k + \beta \sum_{j \in J} w_j C_j$

Constraints:

$$\sum_{o \in O} \sum_{k \in K} \sum_{f \in F_i} X_{ifk}^{o} = 1 \ \forall i \tag{1}$$

$$\sum_{i \in N} \sum_{f \in F} X_{ifk}^{o} + \sum_{f \in F} Y_{fk}^{o} \leq 1 \ \forall o, \forall k \tag{2}$$

$$\sum_{f \in F} Y_{fk}^{1} = 1 \ \forall k \tag{3}$$

$$L_{k}^{0} \leq L_{k}^{max} \forall o, \forall k \tag{4}$$

$$L_k^0 \geq L_k^{0-1} + \sum_{i \,\epsilon\, N} \sum_{f \,\epsilon\, F_i} l_i X_{ifk}^o - \sum_{f \,\epsilon\, F} Y_{fk}^o L_k^{max} \; \forall o > 1, \forall k \tag{5}$$

$$\sum_{i \,\epsilon\, N} X_{ifk}^o \leq \sum_{i \,\epsilon\, N} X_{ifk}^{o-1} + Y_{fk}^{o-1} \; \forall o > 1, \forall f, \forall k \tag{6}$$

$$R_k^o \geq \sum_{i \,\epsilon\, N} \sum_{f \,\epsilon\, F_i} r_i X_{ifk}^{o+1} - \sum_{f} Y_{fk}^{o+1} R^{max} \;\; \forall o, \forall k \tag{7}$$

$$R_k^o \geq R_k^{o+1} - \sum_{f} Y_{fk}^{o+1} R^{max} \;\; \forall o, \forall k \tag{8}$$

$$S_k^0 \geq S_k^{0-1} + \sum_{i \,\epsilon\, N} \sum_{f \,\epsilon\, F_i} p_i X_{ifk}^{o-1} + \sum_{f \,\epsilon\, F} Y_{fk}^{o-1} \, s_f \; \forall o, \forall k \tag{9}$$

$$S_k^0 \geq B_k \; \forall o, \forall k \tag{10}$$

$$S_k^0 \geq R_k^o \; \forall o, \forall k \tag{11}$$

$$C_i \geq S_k^o + p_i - (1 - \sum_{f \epsilon\, F_i} X_{ifk}^o) M \; \forall i, \forall o, \forall k \,\epsilon\, K_i \tag{12}$$

$$C_i \leq S_k^o + p_i + (1 - \sum_{f \epsilon\, F_i} X_{ifk}^o) M \; \forall i \; with \; |P_i|, \forall o, \forall k \,\epsilon\, K_i \tag{13}$$

$$C_i - p_i \geq C_{i'} \; \forall i, \forall i' \,\epsilon\, P_i \tag{14}$$

$$C_j \geq C_i \; \forall i, \forall j \,\epsilon\, J_i \tag{15}$$

$$T_j \geq C_j \; - \; d_j \forall i \tag{16}$$

$$T_k \geq S_k^{|O|} + \sum_{i \,\epsilon\, N} \sum_{f \,\epsilon\, F_i} p_i X_{ifk}^{|O|} - F_k \; \forall k \tag{17}$$

$$X_{ifk}^o \,\epsilon\, \{0, 1\} \; \forall i, \forall o, \forall k \,\epsilon\, K_i, \forall f \,\epsilon\, F_i \tag{18}$$

$$Y_{fk}^o \,\epsilon\, \{0, 1\} \; \forall o, \forall k, \forall f \tag{19}$$

$$C_i \geq 0, T_i \geq 0 \; \forall i \tag{20}$$

$$C_j \geq 0 \;\; \forall j \tag{21}$$

$$T_k \geq 0 \;\; \forall k \tag{22}$$

$$S_k^0 \geq 0, R_k^0 \geq 0, L_k^0 \geq 0 \; \forall k, \forall o \tag{23}$$

Here we propose a non-regular objective function. Its aim is to minimize three terms:

- Tardiness of a well,
- Tardiness of a vessel,
- Completion time of a well.

Constraint (1) ensures that each operation will be scheduled for some vessel. Constraint (2) ensures that each machine is only available to process one operation or one configuration per order. Constraint (3) obligates each vessel to begin with a setup. The need for the load accumulated in an order must be less than or equal to the maximum load limit of the vessel, what is established in (4). Constraint (5) ensures that the total cumulative load of an order is greater than or equal to the load of the previous order plus the loading of the operation in this order. If it is setup, the load will assume zero. Constraint (6) assumes that the operation of a family must be scheduled after another operation of the same family or setup of that family. In (7) the release of an order must carry out the release of the operation at the next order. If there is a setup in the next order, the release should be greater than zero. In (8) the release of an order must be greater than the release of the next order. If there is a setup in the next order, the release should be greater than zero. In (9) the start time of an order is greater than the start time of the previous order plus the processing time of the operation or the setup time in that order. Constraint (10) ensures that the start time of an order must be greater than the initial window of the vessel. In (11) the start time of an order must also be greater than the order release date. In (12) the completion time of an operation is based on the start of the order plus its duration. Constraint (13) ensures that the completion time must be less than its start plus its duration if this operation has any precedence operation. In (14) An operation can only begin if its precedents are completed. In (15) the completion time of a job is the longest completion time between the operations associated with it. In (16) the tardiness of a well is greater than its completion time, less than its due date. In (17) the tardiness of a vessel is greater than the end of its last order, less than its upper window.

The solution to such problem represents vessels voyages which consist of a sequence of activities which have a specified order. After building a schedule of PLSVs activities the Fleet

Sizing Problem can be solved with simulating the schedule with different values of deviations of operations durations. If the simulation model output results show the absence of significant delays for such fleet size, then the schedule can be redone for a lower number of vessels and then the simulation model is run again for a new schedule. This process repeats until the simulation output results show any presence of significant delays.

3 LITERATURE REVIEW AND METHODOLOGY

There are several articles in the considered area that are relevant to routing and scheduling problems that address the fleet sizing problems within deterministic contexts. According to Christiansen et al. (2004), the most important strategic planning problem for all shipping segments (industrial, tramp, and liner) is fleet sizing and composition. Many uncertainties influence the quality of decisions regarding this aspect. There are several major reasons for this uncertainty:

• The long-time horizon that these decisions span, which can reach some years. Sometimes it may span up to 20 and even 30 years when new ships are required in the decision.

• Derived demand for shipping. It depends on the level of economic activity, prices of commodities.

• There is a significant time difference between changes in demand for maritime transportation and the corresponding adjustments in capacities of such services.

Currently, there is no literature offering fleet sizing for PLSVs with stochastic factors influence and schedule determination at the same time. We divide the review into the next parts:

- PLSV scheduling
- Parallel machine scheduling
- Discrete-event simulation modelling

3.1 PLSV scheduling

Queiroz & Mendes (2012) presented *GRASP (Greedy Randomized Adaptive Search Procedure)* metaheuristic with the Path Relinking addition afterwards for solving Pipe Lay Fleet Scheduling Problem. The problem is assumed as a variation of the unrelated parallel machine scheduling problem. The authors developed the metaheuristic, and they mentioned about the advantage of using various ordering rules in the construction phase of the metaheuristic in comparison to using a fixed rule. The proposed metaheuristic didn't concern families of jobs to be scheduled.

There are some recent publications within the problem under consideration. So, Cunha et al. (2018) address the PLSV Rescheduling Problem as the Parallel Machine Scheduling Problem. The mathematical model and the *ILS (iterated local search)* metaheuristic were presented to solve the machine scheduling problem. Both methods were applied to ten instances based on real data. The main measures of efficiency were the average percentage of improvement over the cost of the initial solution. For all the experiments solutions achieved significant improvements. The disadvantage of this method lies in need of having an initial schedule of vessels voyages while in our research we assume that we do not have any schedules in advance. It made us search for the algorithms which could help us to build an initial schedule. For that reason, we referred to the Parallel Machine Scheduling Problem in general case so as to adjust the algorithms solving it to the PLSV Scheduling Problem with unknown fleet size.

3.2 Parallel machine scheduling

The influence of the due dates and the relations between the assignment and the priority dispatching rules is observed by Baker & Bertrand (1981). They introduce the study of a single-machine scheduling problem. The authors admit the significance of meeting of the due dates in production control problems. Therefore, the researchers describe the next two extremes. The first extreme assumes the due date parameters as external data as a declaration of the problem under consideration (the outer extreme). The second one discussing in the article is the case in which the developed system predetermines the due date parameters itself (the inner extreme). The authors consider three assignment rules and five simple dispatching rules that utilize only job-related data as well. The authors analyzed all possible variations of these rules. Therefore, only two assignment rules and two dispatching rules had been chosen for further report. The process of data generation is well-described with some notes about the impact of each value and their distributions. A simple two-level model for a production control system was developed. The main emphasis of rules the authors tested is discussing in conclusion. The work presented by Baker & Bertrand was continued to define a dynamic priority rule with extensions.

The most comprehensive study of the publications, methods, heuristics of the classical *Parallel Machine Scheduling* (*PMS* or $P2//C_{\max}$ as its mathematical notation) problem is a research made by Mokotoff (2001). The survey includes almost all publication from the second half of the 20th century. A complete and quite simple definition of the PMS problem is given. The author noted that lots of production and even life problems could be explained and then modelled as a PMS problem. Therefore, knowing the methods of solution of the problem will leave help to minimize the investigation time. Mostly, there is a research of the scheduling problems that require the exponential time for execution ($NP - hard$), because there are few PMS problems are solvable in a polynomial amount of computation time (P).

At the beginning of the survey, the author defines the notations and mathematical formulations of the original PMS problem. Then the author admits that there are many performance criteria to be considered while solving the problem, and some other notations are presented. In order to deal with the only one and complete model, the author defines assumptions which strictly characterize the classical PMS. As a survey was presented as definitive, it should be proved that the defined problem belongs to *NP-hard* problems. Mokotoff noted that it is possible to verify that $P2//C_{\max}$ is $NP - hard$ by reduction to the partition set problem (Lenstra & Rinnoy Kan, 1979). Therefore, there are not many ways to deal with this class of problems.

The first way of the solution that the author describes in the article is polynomial algorithms that can find the solution for some exceptional cases of the problem. The most common and vital cases that have been studied earlier are summarized and classified by the author. So, there are such classes as identical, uniform, unrelated machines, which in turn divide into independent and dependent primitive and non-primitive jobs. The most important properties of each type are discussed.

On the other hand, it is possible to find the enumerative algorithm to solve the PMS problem. So, the branch and bounds method and dynamic programming might be applied. Therefore, the author formulates a mixed integer programming model to describe the $P2//C_{max}$. It is worth noting that there a quite lot of approaches are developed, but the computational time is still the most blocking value.

The most extensive way to solve the PMS problem is using an approximate algorithm - heuristic. It is important to print out that, even when dealing with approximation algorithms, in some $NP - hard$ problems it could itself be $NP - hard$ to obtain near optimal solution (Arora and Lund, 1996). Therefore, the author starts the description with dividing such algorithms into two types. The generalization of improvement algorithms (such as Tabu Search, Genetic Algorithms) for both identical and unrelated parallel machines is provided. The point is that the Tabu Search became very effective to find a near-optimal solution. The next type of such algorithms are constructive algorithms. It is important to point here, that the most important constructive algorithms dedicated to the PMS problem can be classified by their design in LS (Graham, 1966) and bin packing (based on the MultiFit algorithm of Coffman et al., 1978). So, the list scheduling problem is fully described with its bound calculations and the improvements within the performance. The best algorithm was chosen and described in more detail. The same way the Bin Packing algorithm representation for the PMS problem is presented for different types of it.

Summing up the survey, the author finds the significance of the problem under consideration as one of the most challenging in Combinatorial Optimization. It is worth considering that the research was done almost twenty years ago, but the importance of this study made a huge role in the evolution of the problem and future researches. The author gave suggestions during the survey.

There are many heuristics developed to solve the parallel machine problem; some of them take into account setup of jobs and other variables, others do not. Some of the heuristics take the form of a dispatching rule which determines the value of a priority index for each job. Dispatching rules are designed so that the priority index for each job can be computed easily using the information available at any time. However, these dispatching rules are myopic and usually yield suboptimal solutions. *The Earliest Due Dates (EDD)* rule and the *Weighted Shortest Processing Time (WSPT)* rule are the simplest and most widely used rules (Lee and Pinedo, 1997).

One of the first and basic composite rules for solving the PMS is the *ATC* dispatching rule (Vepsalainen & Morton, 1987). The formula of the rule consists of three well-known heuristics (1). The first multiplier presents the WSPT heuristic, the numerator of the exponent function is an *MS* algorithm. The $\bar{p}$ factor is a mean of the remaining jobs. The k factor here is a due-date scaling parameter, so-called look-ahead parameter. It can be calculated to considerations (2), where R is a relative difference between maximum and minimum due-dates and maximum completion time. Easy to notice, that if k is large enough, the *ATC* rule reduces to *WSPT*, if the parameter small and there are no overdue jobs, the algorithm reduces to *MS*. In our problem, all the tests were with k equal to 0.5. Another special case here is when k is small and overdue jobs exist, the algorithm reduces to *WSPT* applied to overdue jobs. The w, p, d values are weight, processing time and due-date of a job j respectively, t is a time value.

$$I_j(t) = \frac{w_j}{p_j} exp\left(-\frac{max(d_j - p_j - t,\ 0)}{k\bar{p}}\right) \quad (1)$$

$$k = 4.5 + R, \text{for } R \leq 0.5 \quad (2)$$

$$k = 6 - 2R, \text{for } R > 0.5$$

Lee & Pinedo (1997) developed their dispatching rule with the post-processing – the simulated annealing procedure. They consider the scheduling problem on parallel machines with total weighted tardiness with completely random setup times. In their work, the authors attempted to estimate the makespan analyzing how factors of statistics for scaling parameters affect it. Ten completely different instances were generated with the help of the uniform distribution. To show the significance of scaling parameters obtained previously, the measure of performance was presented. So, the relative error and the normalized relative error were computed, and the result table was presented. It is worth noting that the authors compared instances with ten jobs and two machines, and they noted the vital significance of the variety of instances to be generated. The simulated annealing procedure is described from multiple sides, such details as nationhood function or searches to reach the preferable state are well described. The average relative improvement is presented, as well. It has to be noted that for that computer powers, there was no efficient algorithm developed for solving the sixty-job problem optimally. The simulated annealing searching procedure becomes negligible when the number of searches overcomes sixty. As a result, the *Apparent Tardiness Cost with Setups (ATCS)* rule was developed.

As we can notice, the parallel machine's interpretation can be applied in different practical cases. So, Pfund & Fowler (2007) applied the approach in the semiconductor industry to achieve the goal of on-time delivery. Basing on *ATCS* rule developed by Lee & Pinedo (1997), the new approach was developed. Considering the *ATCS* approach was implicitly affected by the ready times, the new exponential term was added. The authors also tried to estimate the impact of ready time r of job k. In opposite to Lee & Pinedo (1997), where the simulated annealing post-processing procedure was used, the authors proposed a grid search method that generates multiple schedules. Completely the same method of instance data generation was used for this approach. The comparison table and analyze of average performance of approaches to base $ATCS$ grid approach are provided. As one of the measurements of the result, the authors got is a collation of computation time and the quality of the solution. In $NP - hard$ problems, this can influence a lot regarding what method to choose, so the authors found that the computation times for EDD and WSPT were approximately 0.002 seconds and computation times for formula approaches were approximately 0.006 seconds, whereas the grid one took more, as expected. As a conclusion of the article, the authors note, that the $ATCSR$ formula approach gives results which are about 9% worse than the $ATCSR$ grid approach. Some theoretical improvements as reducing of the size of the grid are provided.

The attempts to improve existing approximate algorithms continue in much relevant present research. Some of them are looking for a way to apply some new constraints in the formula. Based on the well-known ATC dispatching rule (we described above), Su et al. (2017) developed a two-phase heuristic for minimizing the total weighted tardiness subject to the machine eligibility constraints. Firstly, the authors introduce us to the classical PMS problem and consider two different fields where the solution to the problem plays a huge role. So, they apply the model to Operating Theatre Scheduling and Container Terminal Crane Scheduling both. Discussing the previous studies, the authors present a two-phase heuristic and provide extensive experimental performance results. Then, Su et al. formulate some assumptions and derivate some interesting special cases if all jobs have a common due date.

Further, based on the found properties, they develop the improvement of the heuristic. The authors devote a section for analyzing the symmetric and asymmetric forms of nestedness of the machines. They study the significant differences between the forms and provide the results and insights. Generally, the developed framework consists of two phases. The first phase is a computational one, where the authors compute the job and machine flexibilities and average machine load factors, estimate them and evaluate the makespan. In the second phase of the describing heuristic, the authors evaluate the proper scaling parameters and present the $ATCF$ dispatching rule. So, they make four general conclusions of the results. It is worth noting that Su et al. use the *Sequential Uniform Design* method in order to obtain the proper pair of such scaling parameters (Fang & Wang, 1993). The usage of this uniform method explains with the ability to obtain the relations between the responses and the factors quickly. After Su et al.

evaluate the performance of the describing dispatching rule, they present a complete comparison with the default *ATC* rule and make an emphasis on providing the explanations. As the authors made some performance evaluations to the *ATCF* rule and got three more rules, they compared them as well. One essential section in the research is testing on the real dataset. The authors collected the data from Shanghai General Hospital, one of the largest hospitals in China, over the period 01/01/2016–06/30/2016. They define a new measure of evaluation – *The Improvement Ration* and provide tables with the derived results. To conclude, the authors admit that the dispatching rule can be improved significantly, and this improvement does not require much additional computational time. There are some notes which could be studied further, as considering setup times explicitly as it was made by Pfund & Fowler (2007), dealing with situations in which jobs are released at different times, explanation of the phenomenon when the job and machine flexibility are high, and the developed rule performs rather good. The last assumption is explained as the objective parameters are close to the line (to the reality). Therefore, it is easy to obtain a proper schedule. The opposite situations are explainable this way. The emphasis of the observing study is the nestedness of the machine eligibility constraints, so probably the further researches may consider the different case – unnested of the constraints.

It is worth noting that PMS problems found in the literature described above, didn't consider families of jobs. But in our research problem, we have different types of operations which have different setup times.

3.3 Discrete-event simulation modelling

Shyshou (2009) developed a discrete-event simulation model for evaluation of alternative anchor handling tug supply (AHTS) fleet size configurations for the Norwegian offshore oil and gas operator. AHTS vessels have their own activities nature and sequence of operations what differs them from PLSV vessels. The author chose Arena as a simulation tool and *SPSS* for simulation output results analysis. He underlines the ability of convenient modelling using Arena's Object Model as it is integrated with Microsoft Excel for reading inputs and writing outputs for further analysis.

The top-level flowchart for the simulation model proposed by Shyshou is depicted in Figure 3.3.1. The sources of uncertainty include durations of anchor handling operations and weather conditions. The author emphasizes that unpredictability of weather makes the presented problem highly stochastic. So in this paper, the operations durations are modelled with the help of data about the weather. In our case, deviations of operations durations can also be affected by bureaucratic procedures, and therefore we need to generalize the rule for changes in durations.

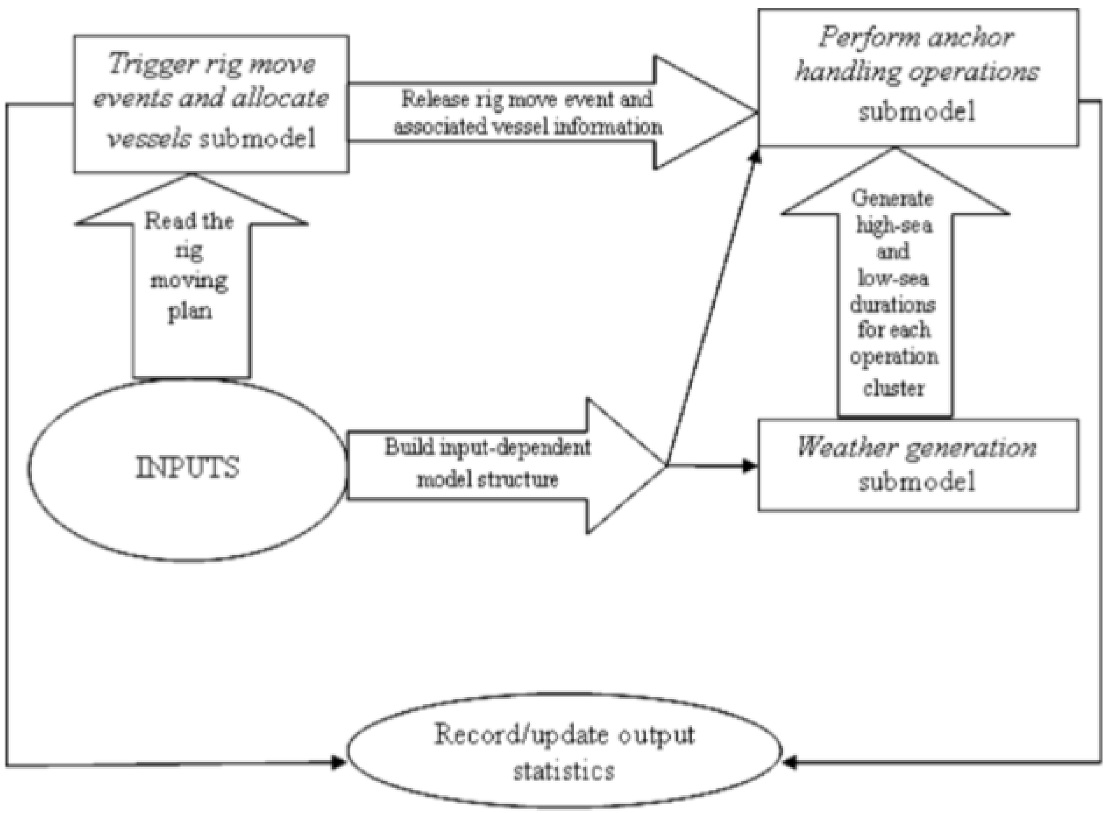

Figure 3.3.1 Arena implementation flowchart proposed by Shyshou (2009)

The author also used Arena's abilities in using animation. It is illustrated in Figure 3.3.2. So it gave him an ability to monitor changes in the number of vessels and operations performed in real time. Creating animation models in Arena requires good knowledge of using this software. Molde University College has a special course dedicated to building discrete-event simulation models which lasts for the whole semester.

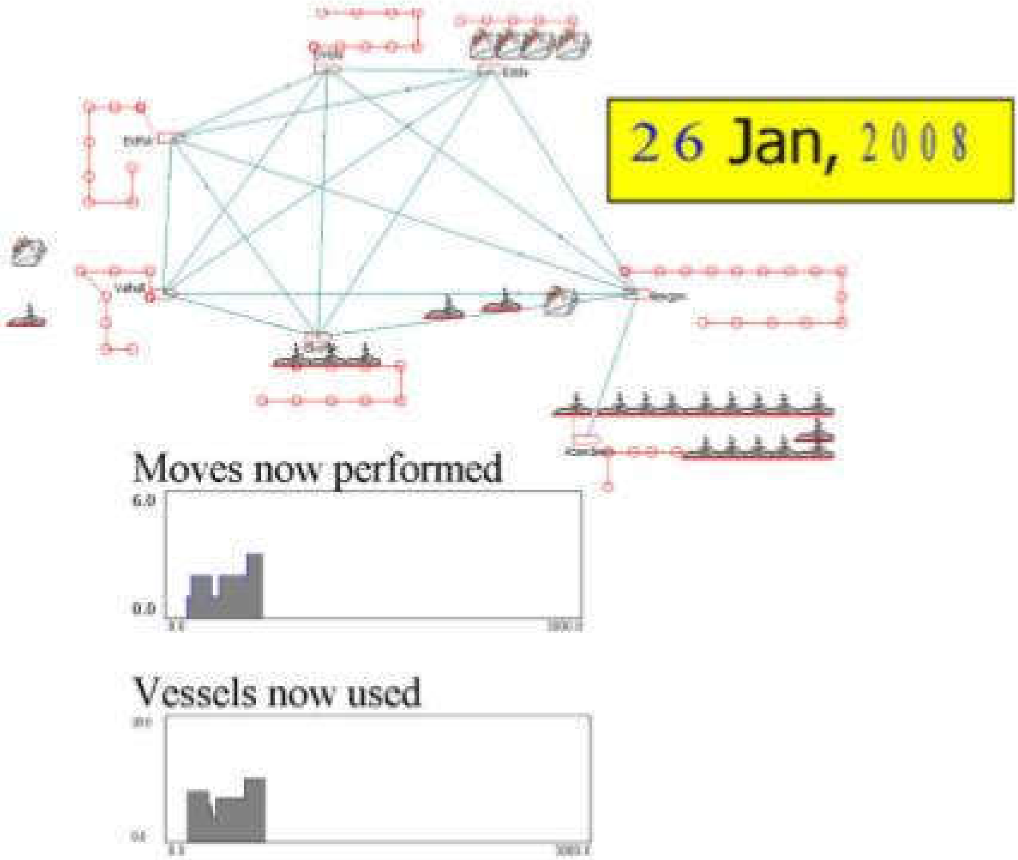

Figure 3.3.2 Animation process developed by Shyshou (2009)

Maisiuk & Gribkovskaia (2014) address a supply vessel planning problem arising in offshore and present a discrete-event simulation model for evaluation of alternative fleet size configurations considering uncertainties in weather conditions and future spot vessel rates. Here a vessel voyage is represented as a set of activities performed sequentially by a supply vessel and viewed as a process evolving. The authors chose Arena as the discrete-event simulation environment and *PASW* Statistics for data analysis. They also mentioned that collecting weather data was a significant issue to be solved. The logic flowchart illustrating the conceptual design of their model is presented in Figure 3.3.3. The model includes the logic of assigning an available time-charter or spot vessel in case of unavailability of a scheduled vessel. In our simulation model we also want to include the logic of assigning an available vessel, but this vessel will be chosen from the common amount of vessels of the same type as we have only one type of vessels – PLSV.

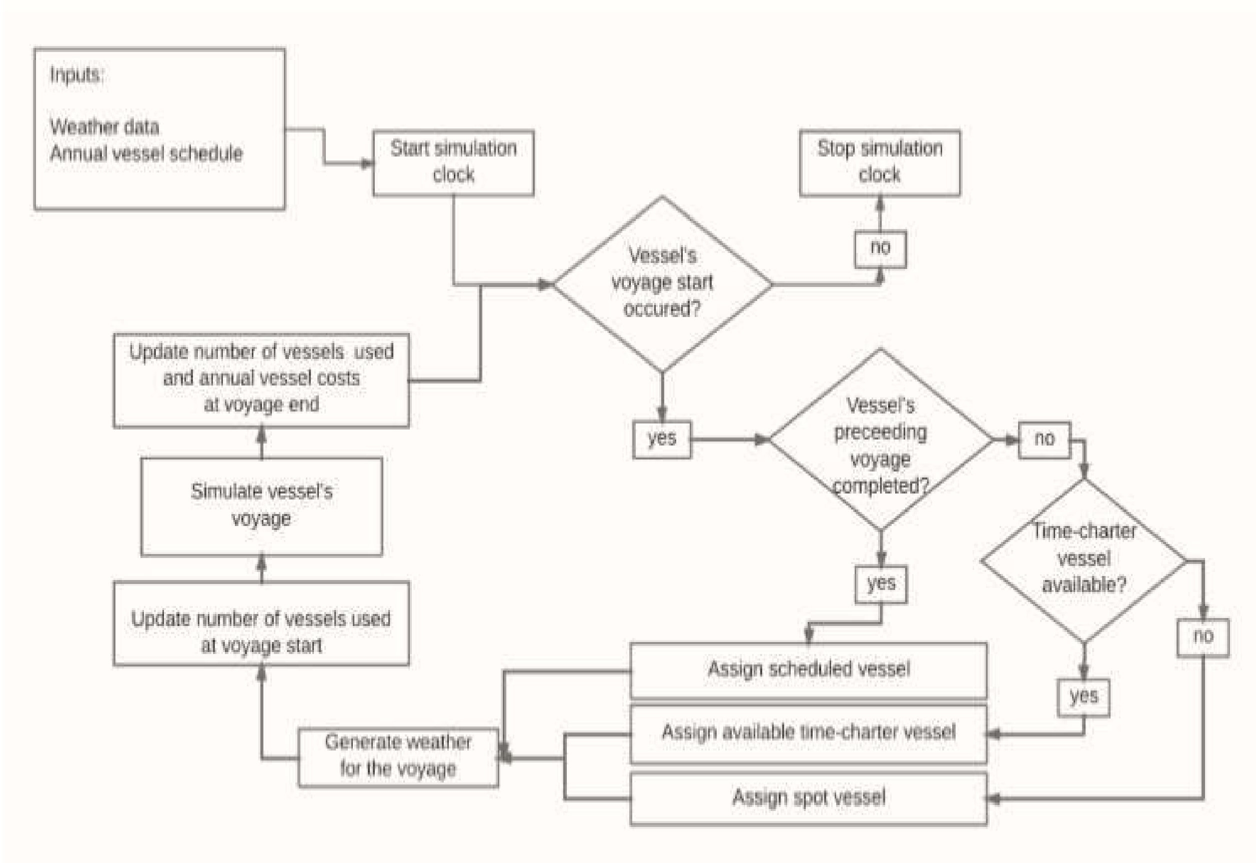

Figure 3.3.3 Logic diagram of the discrete-event simulation model proposed by Maisiuk & Gribkovskaia (2014)

4 DATA AND DECISION SUPPORT TOOL IMPLEMENTATION

This paper implies creating two projects: one for building a schedule and another one for building a simulation model. The first project accepts input data in the form of instance files with extension *.txt*, processes it and creates input data for the second project, including the whole information about the schedule:

1. the initial amount of vessels
2. amount of operations
3. what operations are performed by each vessel
4. start time for each operation
5. planned duration for each operation
6. whether an operation needs setup before it, i.e. whether it's first in the vessel voyage
7. the due date for each operation
8. setup duration for each operation
9. time windows of availability for each vessel

It was decided to implement building a schedule with the help of such technologies as C++ using its development environment Microsoft Visual Studio 2017. The simulation model was created in Arena 15.0, the discrete-event simulation environment software from Rockwell Automation Technologies, Inc., as it provides high-level simulation models development with abilities of reading data from files, visualization of simulated processes and preparing well-readable reports.

It's very important to notice that the simulation model will contain the logic of assigning available vessels in real time to those operations which could not be processed by their initial vessels at their scheduled start times. Available vessels here are considered to be waiting to implement their next voyage and have a suitable time window till it so as to pick up the operation which needs processing, i.e. this time window should place setup duration of the operation and its duration. This idea comes from the fact that if the company management encounters that an operation in the schedule (which does not assume any delays in it) cannot be processed at its start time, then the management can immediately assign an available vessel to it so it could meet its start time. As the schedule does not contain delays initially, following the schedule as much as possible reduces the risk of delays occurrence.

The following scheme illustrated in Figure 4.1 shows one iteration flow of data from an instance file to the output result, which shows possible delays and the number of rescheduled vessels and total completion time.

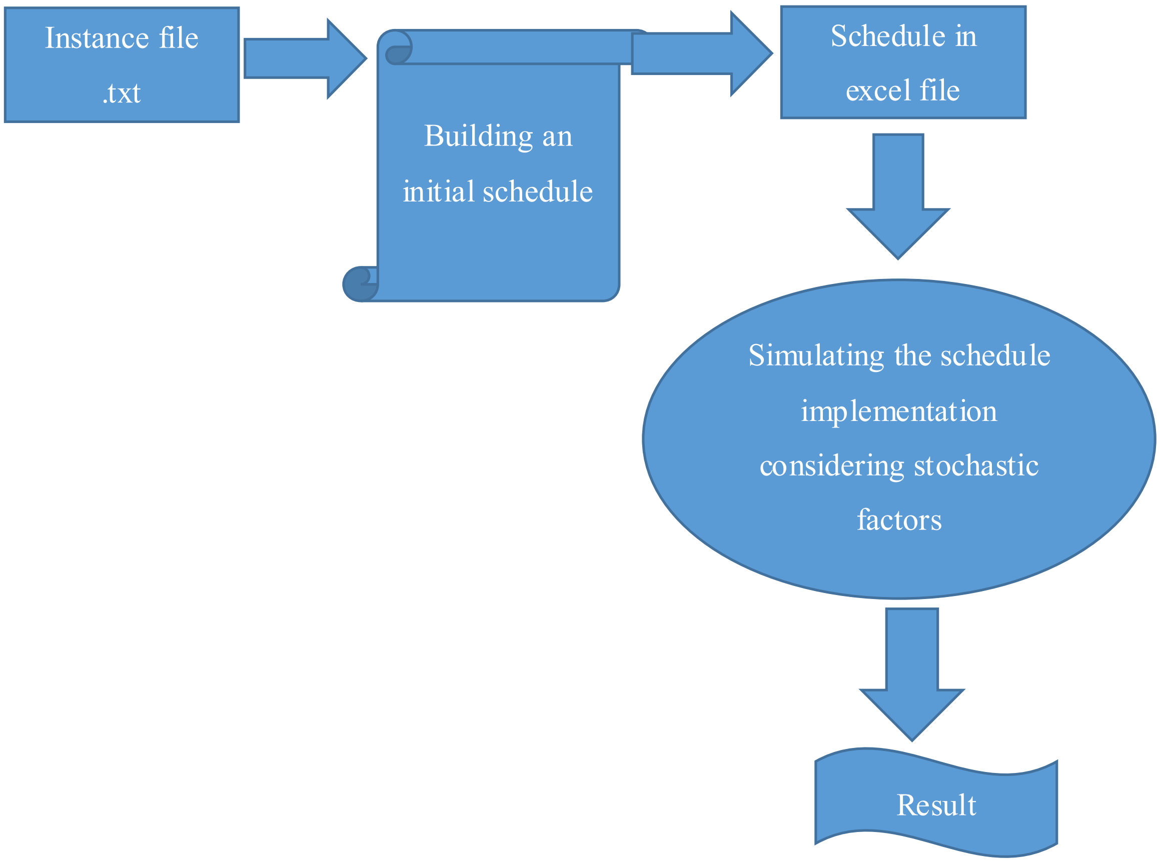

Figure 4.1 The data flow from an instance file to the output result

4.1 Data

The input data for building an initial schedule of PLSV voyages for the first project implementing the optimization methodology was formed into real-based instance files which were organized as follows:

number of operations (n) – INTEGER

number of wells (w) – INTEGER

number of types of operations (f) - INTEGER

p - processing times of operations - VECTOR

r- release dates of operations - VECTOR

l - load occupation rate of operations - VECTOR

q - family of operations - VECTOR

w - weight of wells - VECTOR

d - deadline of wells - VECTOR

s - setup time durations of families - VECTOR

$(a_{ik})_{n \times w}$ - operations and wells association (1 if operation i is associated with well k and 0, otherwise) – MATRIX

4.2 Decision support tool implementation

The methods described below imply using modern and effective technologies which pass data from one to another what makes the process of gathering results fully automated and therefore less risky to be wrong as the role of a human in passing data here is minimal. This fact becomes very important when the number of vessels and operations increases. Avoiding the hard code of data in the simulation model makes it very flexible and reusable, better modifiable, makes it easy to be run on many different schedules with different amounts of vessels and operations. And therefore the iterative process of searching for a suitable fleet size became possible.

4.2.1 Algorithm for construction of PLSV schedule

An initial schedule of vessels is built with the help of the proposed constructive heuristic based on the ATC algorithm adjusted to a real PLSV Scheduling Problem. Its pseudo-code is the following:

```
Given(
   J: Wells
   N: Operations
   K: Machines
   F: Families
   pi: processing times of operations
```

```
    r_i: release date of operation
    l_i: load occupation rate of operation
    f_i: family of operation
    w_i: weight of well
    s_q: setup time duration for family
    d_j: due date of well j.
    X_ij: operations x wells association
) Start:
MachineCompletion = M.size
Solution = M.size
For(k ∈ M) Do
    MachineCompletion[k] = k.release
    Solution[k].machine = k
End For
For(i ∈ N) Do
    For(j ∈ J) Do
        TaskWeight[i] = j.weight
    End For
End For
For(q ∈ F) Do
    AverageSetup += q.setup
End For
AverageSetup /= F.size
While(true) Do
    time = Infinity
    For(k ∈ M) Do
        If(time > MachineCompletion[k])
```

```
            time = MachineCompletion[k]
         End If
      End For
      total = 0
      AverageDuration = 0
      For(i ∈ N) Do
         If(!marked[j])
            Calculate(FullDuration, AverageDuration, Total)
         End If
      End For
      AverageDuration/= total
      BestTask = ATC()
      BestMachine = -1
      If(!BestTask)
         Break
      End If
      Chosen = N[BestTask]
      For(k ∈ M) Do
         MachineAllowed(chosen)
         TestBatch(chosen)
         LimitExceeded(chosen)
         TestBatchFamily(chosen)
         CalculateShiftForward(chosen)
         CalculateNewDelays(chosen)
         Return BestMachine
      End For
      If(!BestMachine)
         Break
```

```
62:     End If
63:     If(SameBatch)
64:        MoveTasks()
65:     Else InsertSetup()
66:     End If
67:     InsertNewTask()
68:     UpdateMachineCompletion()
69:     marked[BestTask]=true
70:  End While
71:  Save Solution
72:  Print Solution
```

The heuristic considers the following input parameters: the set *J* of wells, the set *M* of machines (vessels), the set *N* of operations (tasks) and the set *F* of families. As the primary problem is the Fleet Sizing Problem, it is unclear what number of vessels to use, so the set of the machines is limited with a large number no less than the amount of operations. The processing times, release dates, load occupation rates and families of operations, weights and due-dates of wells, setup durations of families and matrix of association between wells and operations are provided as well. At lines 15 and 16, the current solution and completion time of each machine are set. The cycle on line 21 sets the tasks weights. Then at line 26, the average setup is calculated. The primary cycle starts at line 30. It begins with calculating the current minimum completion time of each machine. Then, at line 39, the cycle calculates the total number of non-scheduled tasks, their durations summarized with the setup durations, average duration. The appliance of *ATC* dispatching rule is seen at line 45, where the appropriate task index is calculated. If no such index is found, the cycle stops. Then, the task with the corresponding index is evaluated and the search for the machine, which will give the minimum delay or the completion time, in the same batch or not (the PLSV voyage) begins at line 51.

The testing of such batches in order to avoid the case when the family of the last setup and the task’s family differ is computed at line 53. Then, the shift of the batch and new delays are calculated. After this, if no machine was chosen, the primary cycle breaks. Otherwise, it selects its schedule. If the same batch, moving of tasks forward is performed, whether necessary. If the different batch, setup is inserted. At the end of the primary cycle, insert of the new task occurs, it becomes marked as visited, machine completion time is updated. After the algorithm finishes, the schedule is evaluated, saved in the *.xlsx* file and printed in the console to observe the results.

4.2.2 Simulation model

The created simulation model has two types of entities representing vessels and operations which they have to implement. These entities are created simultaneously with the help of module Create at the first arrival, and their number is specified in an input excel file (Figure 4.2.1). The entities continue their life-cycle specified by the model *in parallel*.

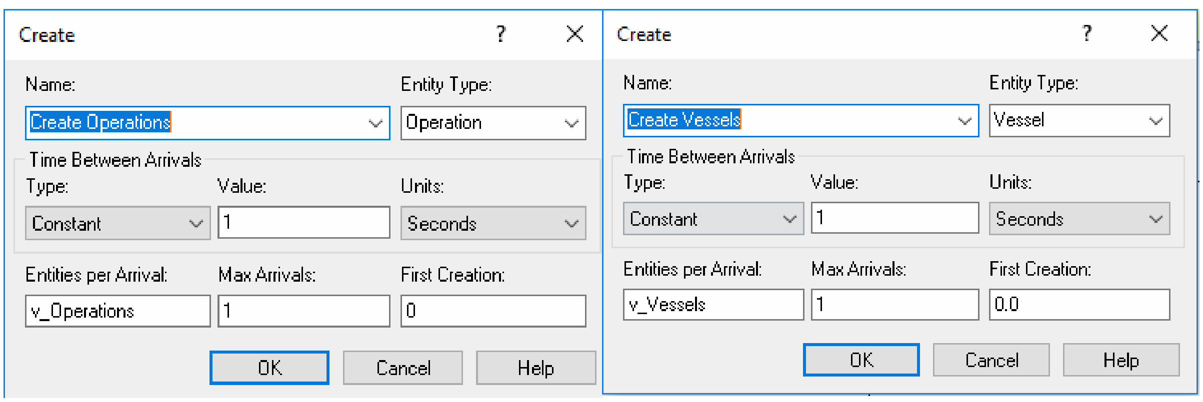

Figure 4.2.1 The *Create* modules

The values specifying the number of vessels and operations are stored in variables *v_Vessels* and *v_Operations* (Figure 4.2.2) directly from the file what makes the creation of entities quite flexible.

Variable - Basic Process

	Name	Comment	Rows	Columns	Data Type	Clear Option	File Name	Recordset	File Read Time	Initial Values	Report Statistics
1	v_Operations				Real	System	File 1	Operations	BeginReplication	0 rows	☐
2	v_Vessels				Real	System	File 1	Vessels	BeginReplication	0 rows	☐

Figure 4.2.2 Variables *v_Vessels* and *v_Operations*

Each of the entities has its own characteristics – attributes. For operation they are its identification number, duration, start time, due date, identification number of the vessel which is initially assigned to implement this operation by the schedule, setup duration. Also it has auxiliary attributes such as a value which shows if the operation needs setup before it, i.e. whether it's first in the voyage or not; stochastic delta value which represents random factors influence and which has certain probability distribution; an attribute which shows if the operation is implemented according to the initial schedule or not. Vessels have such attributes as a vessel identification number, time windows of availability and number of operations which are implemented by them according to the initial plan.

The model segment responsible for creating all entities and initializing their attributes is illustrated in Figure 4.2.3. The ReadWrite module was used to read attribute values from the recordset sequentially and assign them to entities. It looks similar for all attributes. Figure 4.2.4 illustrates it for an attribute *operation_id*. The recordset *Durations* stores values for operation ids from the excel input file.

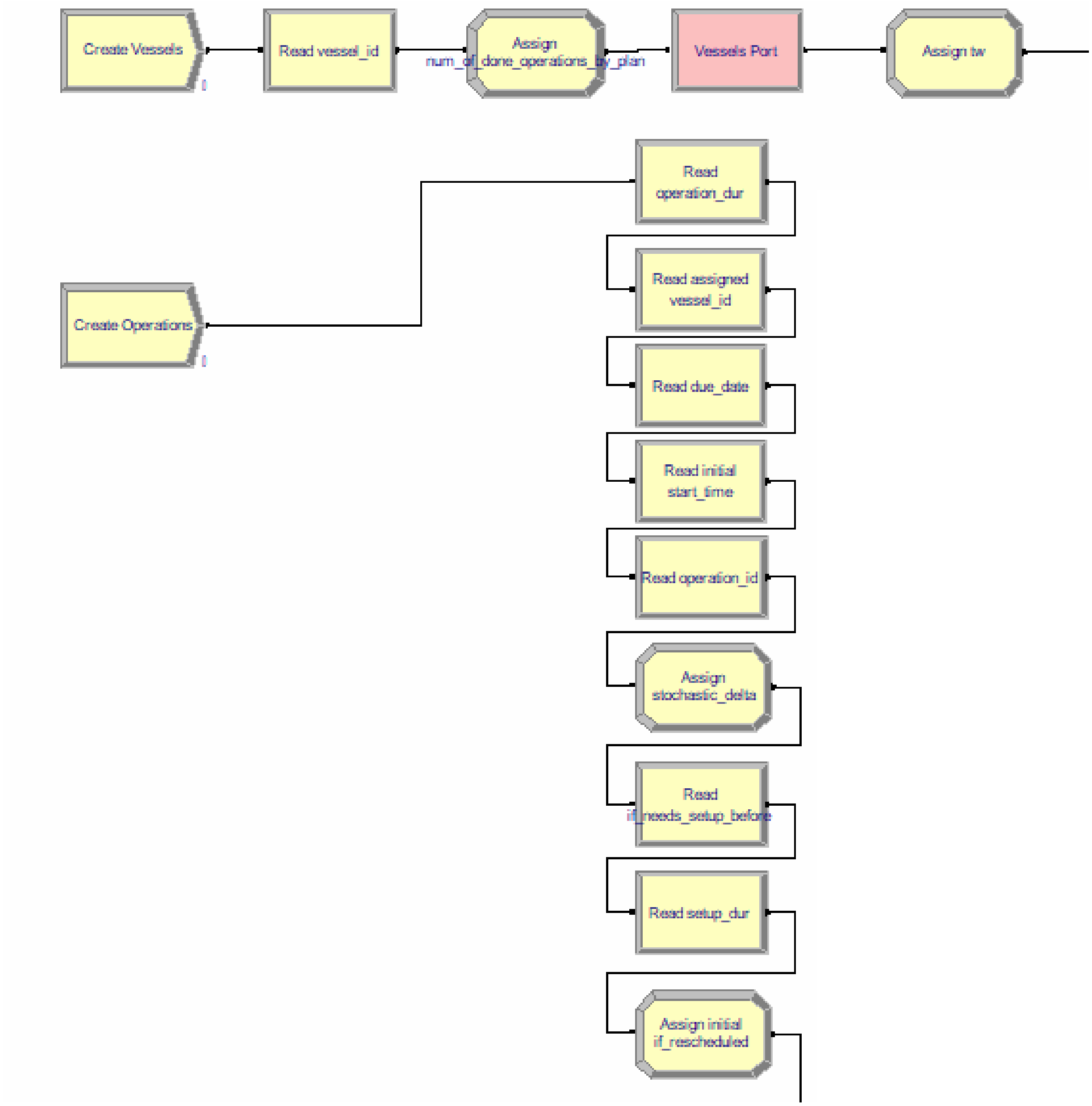

Figure 4.2.3 The simulation model segment responsible for creating entities and initializing its attributes

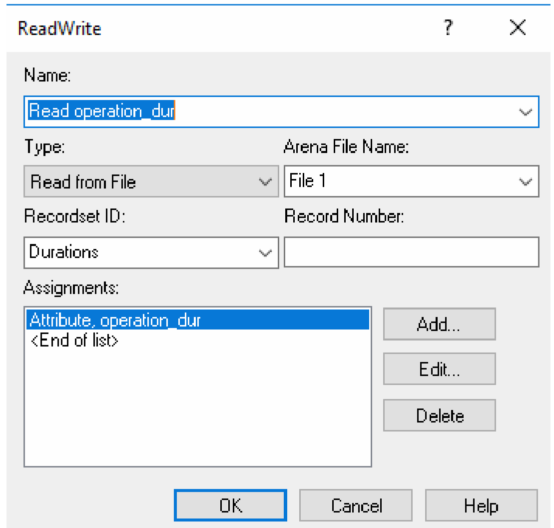

Figure 4.2.4 Reading *operation_dur* attribute from the file

The Station module *Vessels Port* was used here as a point for vessels to return after it implements an operation and as all vessels should have their availability time windows updated after each operation implementation, the module of assigning an attribute *tw* responsible for availability time window goes after the module *Vessels Port*.

The module segment responsible for assigning a vessel to an operation is illustrated in Figure 4.2.5. So as to put them into the time system according to the initial schedule and simulate a real-time process Wait modules are used: *Wait till start time of setup* and *Wait till start time*. As an operation can be set in a voyage with other operations it is needed to check if the start of processing it will begin at its scheduled start time or the start time of its setup. That is why the module Decide called *If needs setup* is used here. The attribute *if_needs_setup_before* is binary therefore the module checks whether its value is equal to 1. If so, then an entity goes into Delay module called *Wait till start time* of setup which holds an entity for a certain amount of time. In this case, this delay is equal to the difference between start time and setup duration as all entities are created at time 0. If *if_needs_setup_before* is equal to 0 then an entity should wait till its start time.

The Search module called *If vessel is available* checks if a vessel which was initially assigned for operation by the schedule is ready to implement it, i.e. has finished its previous operation in time and is waiting to implement its next operation. The internal part of this module is depicted in Figure 4.2.6.

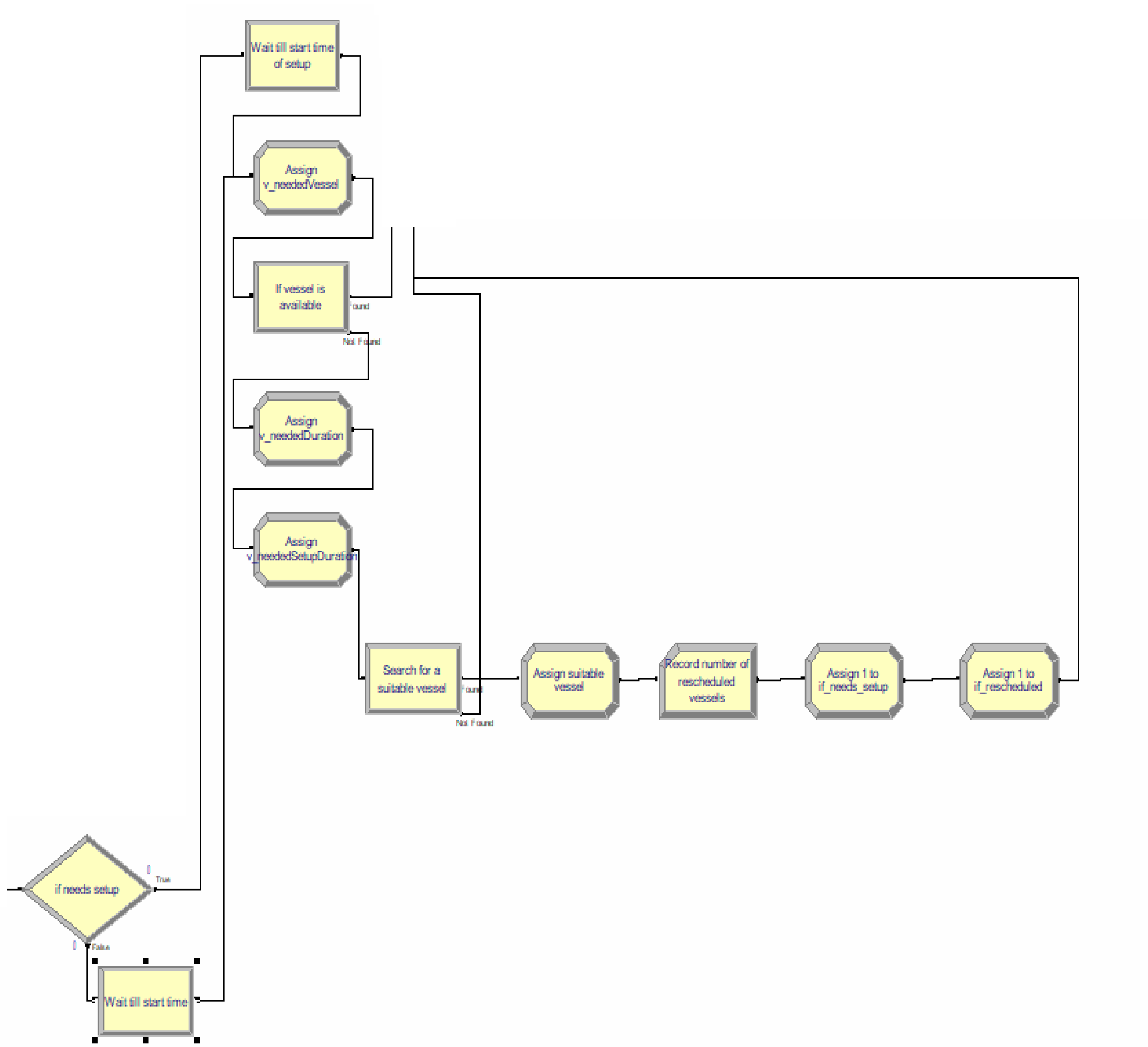

Figure 4.2.5 The simulation model segment responsible for assigning a vessel to an operation

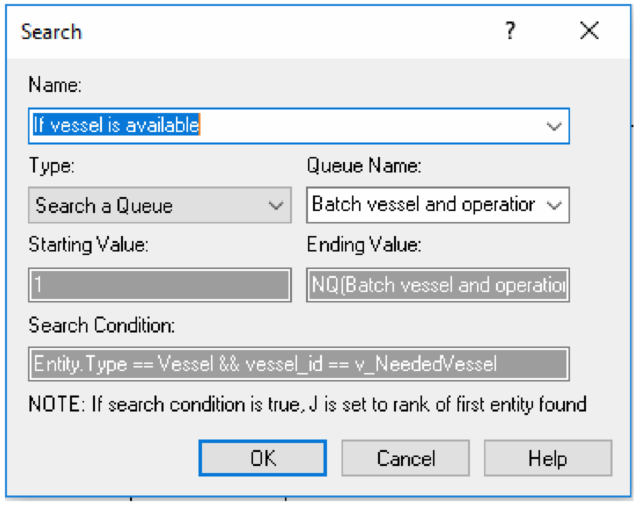

Figure 4.2.6 The Search module for checking availability of an assigned vessel

The module implements search in a queue where vessels get after visiting *Vessels Port.* This queue is called *Batch vessel and operation*. The module starts from the first its element and processes it till its last element as the function NQ returns the size of the specified queue. The processing takes place until the condition that the element is a vessel with a needed *vessel_id* is met. If the needed vessel is available, the operation goes up and forms a batch with that vessel. Otherwise, it's needed to reassign some available vessel to it. For that purpose, the Search module *Search for a suitable vessel* is made. Its content is depicted in Figure 4.2.7. The module returns the position of the first found element satisfying the condition.

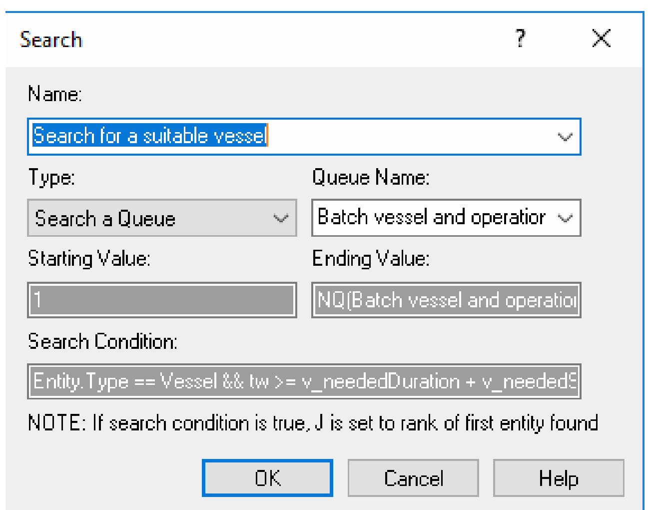

Figure 4.2.7 The Search module for searching for a suitable available vessel

The condition implies that a vessel should have an availability time window which can place both the duration of an operation and setup for it. As a global variable *J* will store the position of the satisfying vessel in a queue it's then needed to retrieve the *vessel_id* which belongs to the found vessel and assign it to the operation attribute *vessel_id*. It's done in the Assign module *Assign suitable vessel* depicted in Figure 4.2.8.

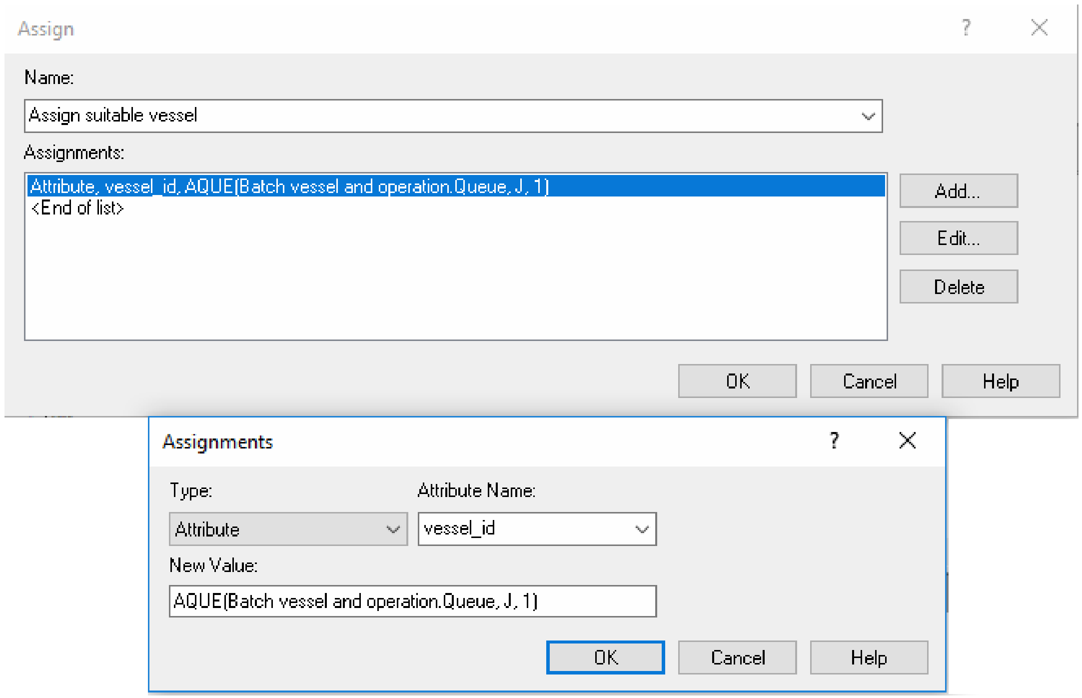

Figure 4.2.8 The Assign module for assigning an available and suitable vessel

The method AQUE here returns the value of the first attribute of the element in the queue, which has the position J in it. This value is then assigned to an operation *vessel_id* attribute.

So as to gather information about rescheduling the Record module *Record number of rescheduled vessels* is applied. It counts the number of vessels which were reassigned for operations which did not meet their initial vessels (Figure 4.2.9).

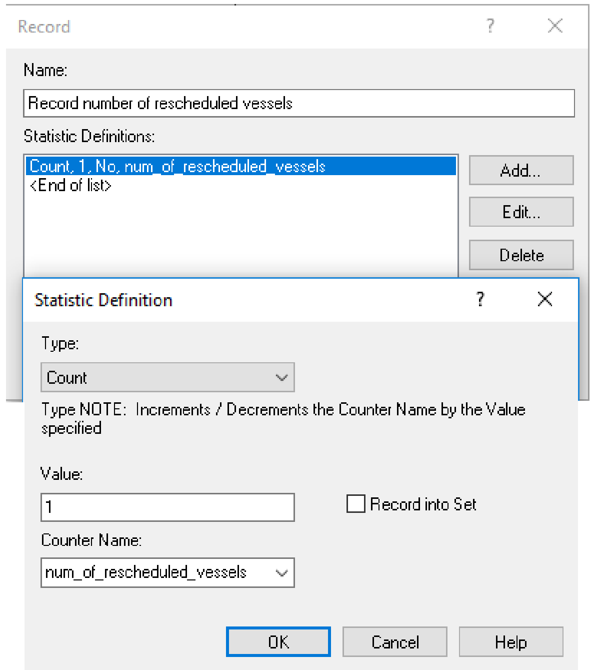

Figure 4.2.9 *Record number of rescheduled vessels* module

This counter is increased by 1 each time an entity passes through it.

The next segment of the proposed simulation model represents the logic of implementing an operation by a vessel. It is illustrated in Figure 4.2.10.

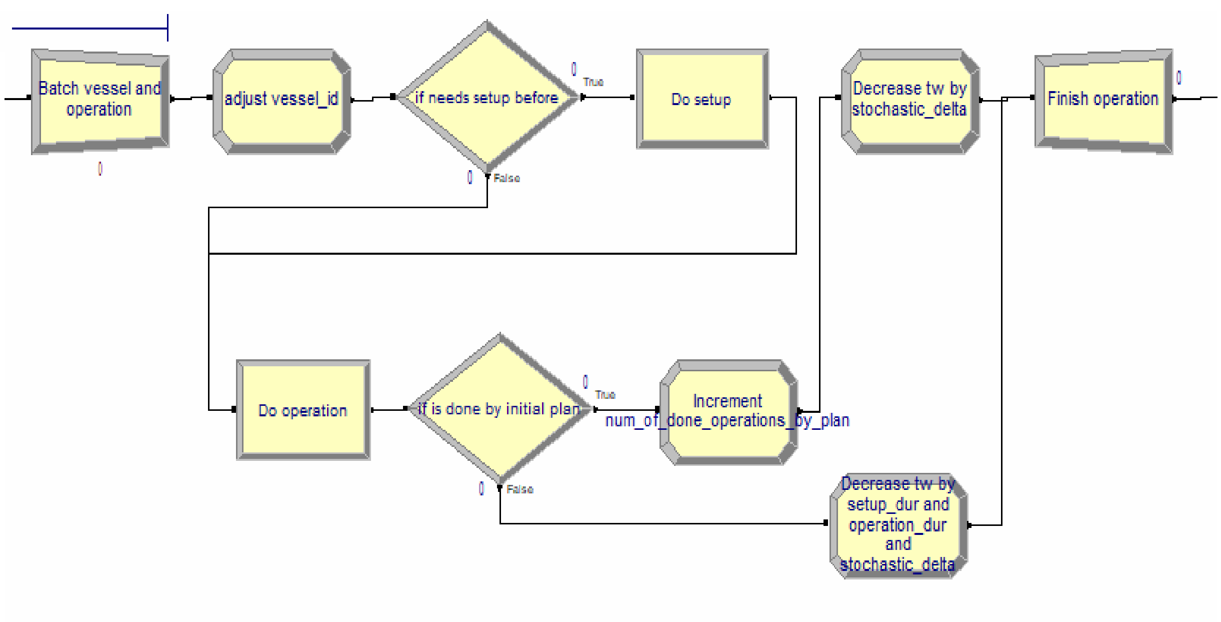

Figure 4.2.10 The simulation model segment representing implementation of an operation by a vessel

As soon as an operation is assigned an available vessel (by initial plan or a reassigned one) it goes to the Batch module called *Batch vessel and operation* which plays the **key role** in the whole simulation module. This module joins two entities, a vessel and an operation, into one temporary entity by an attribute *vessel_id* (it means the vessel and the operation coming into the queue should have an equal *vessel_id* so as to be joined) with all attributes which belong to the vessel and operation and makes this entity passes the next path as a whole one till it's split. The Batch module content is showed in Figure 4.2.11. The batch size 2 means that the module batches two entities into one. The Save Criterion here is Sum what means that all attributes of two batched entities will be summarized for a new entity what does not corrupt the values of its attributes as the attributes of a vessel and an operation intersect only in *vessel_id* which will be adjusted in the next module by dividing it by 2.

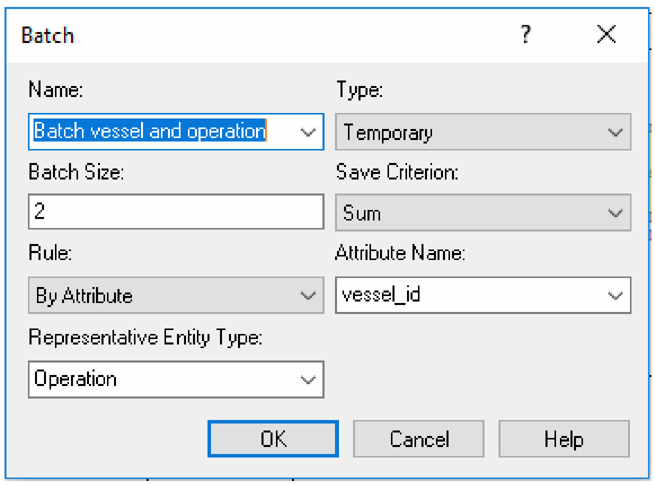

Figure 4.2.11 The Batch module content

As soon as a combined entity is created it passes through the Delay modules depending on the value of the attribute *if_needs_setup*. If it's equal to 1 then before entering the Delay module called *Do operation* it is delayed in the module *Do setup*. In this model segment, the influence of stochastic factors during implementation of operations is taken into consideration as it's shown in Figure 4.2.12.

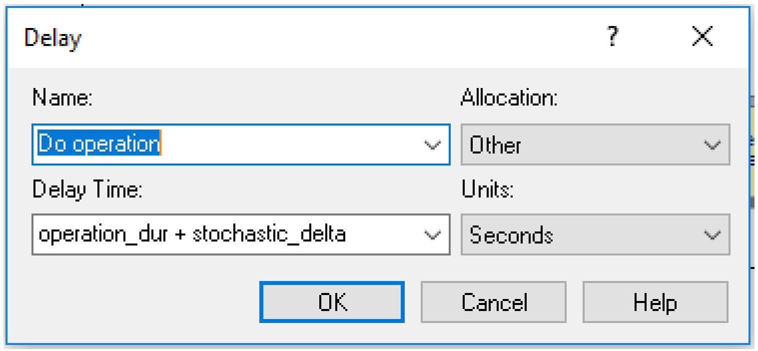

Figure 4.2.12 The Delay module *Do operation*

The *stochastic_delta* here plays the role of a random deviation of operation duration. We assume it to have a triangular distribution TRIA(0, 5, 10) with a minimal value equal to 0 days, 5 days in average and 10 days in maximum as it is considered to be the most suitable distribution describing our case. Anyways, if researchers decide to introduce another distribution for operations durations deviations, it can be easily replaced.

Every time a vessel implements an operation it is needed to update its time window of availability as it becomes less due to stochastic factors influence. A variable which stores all values of availability time windows for all vessels is a global 2D variable *v_TW*. A row *i* of this matrix corresponds to a vessel with *vessel_id = i* and a column *j* reflects a time window between operations *j-1* and *j* which should be done by the vessel according to the initial plan. Current time window of a vessel is a cell of such a matrix and is stored in an attribute *tw*. So as to update a proper cell of the matrix it is needed to monitor the number of done operations by the plan of each vessel. If some vessel is reassigned for an operation this attribute value will not be increased by 1 and its current time window stored in *tw* attribute will be updated but not the next one, and for the next operation this time window will be used. Also, it is important to notice that in this case, the time window should be decreased not only by the value of stochastic factors but also by the operation duration and the setup duration. This decrease is implemented in the Assign module *Decrease tw by setup_dur and operation_dur and stochastic_delta,* which content is illustrated in Figure 4.2.13.

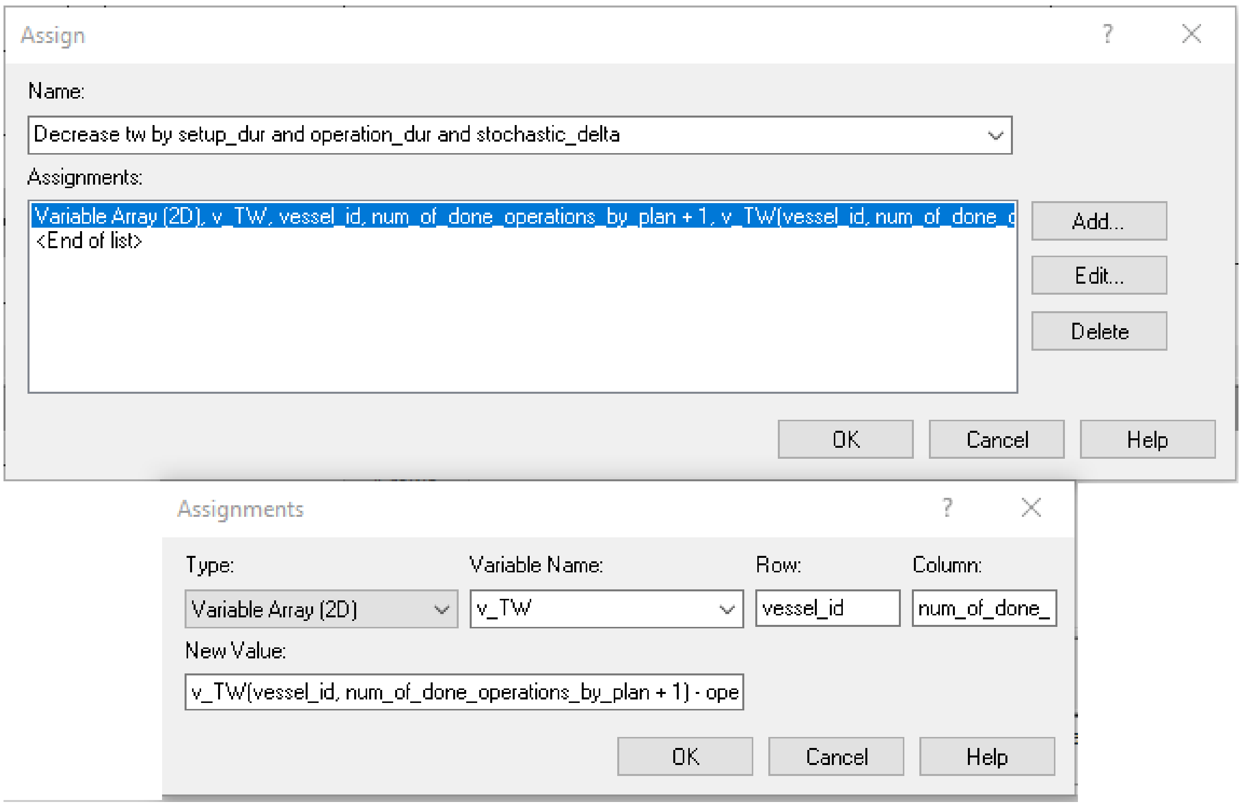

Figure 4.2.13 The content of the module responsible for updating a time window of availability of a vessel

But if a vessel implements an operation by the initial schedule then after implementation the next time window will be updated and assigned to the vessel. This case is processed in the Assign module *Decrease tw by stochastic_delta*.

As soon as an operation is done by a vessel, a temporary batched entity should be separated so as to make the vessel do its next operation and gather information about the operation completion time. For that purpose, the Separate module *Finish operation* is used. It is illustrated in Figure 4.2.14.

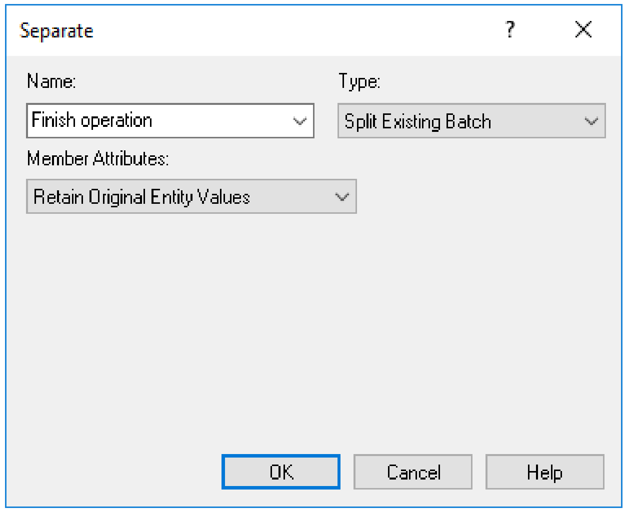

Figure 4.2.14 The Separate module *Finish operation*

The following simulation model segment is depicted in Figure 4.2.15. Its main logic lies in the fact that an entity representing a completed operation should be disposed and an entity representing a vessel which made that operation should get ready to do the next operation.

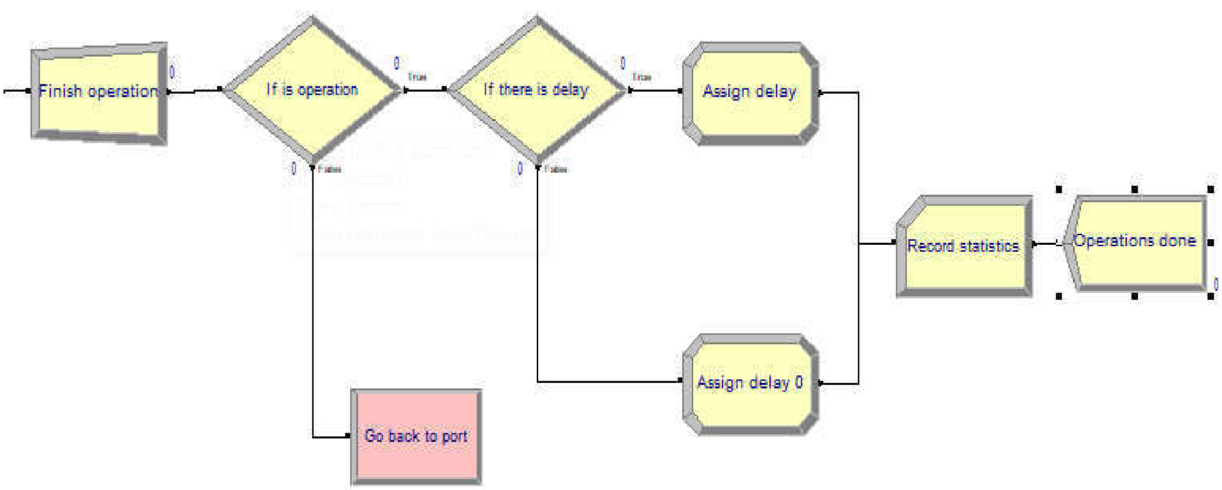

Figure 4.2.15 The simulation model segment representing the completion of an operation by a vessel

Even though the whole vessel's current voyage may not be completed yet and the vessel may not have to go back to the real port, using the Route module *Go back to port* here (Figure 4.2.16) does not violate the common logic of sequential implementation of operations in one voyage by a vessel. As a route time here is considered to be 0 and setup implementation is regulated by the attributes *if_needs_setup* and *setup_dur* in the simulation model segment representing the implementation of an operation by a vessel in Figure 4.2.10.

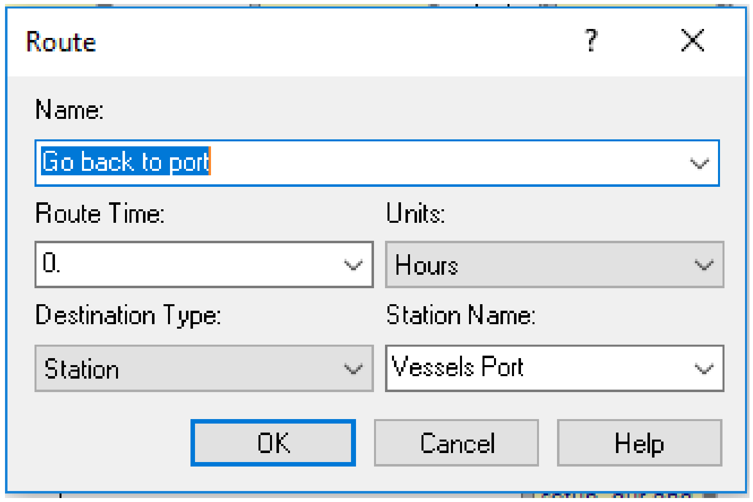

Figure 4.2.16 The Route module *Go back to port*

As after the batched entity is separated the entities of a vessel and operation have two different paths, it is possible to distinguish these paths with the help of the Decide module *If is operation* which content is depicted in Figure 4.2.17.

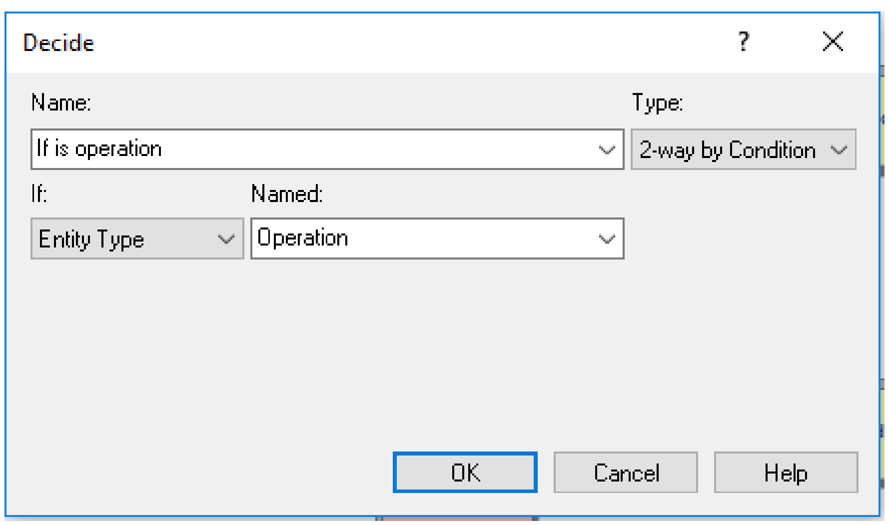

Figure 4.2.17 The Decide module *If is operation*

So if the coming entity represents an operation its further path will lie on the branch representing the true condition. Otherwise, as we have only two types of entities, the entity will represent a vessel and will go through the route *Go back to port*.

Before being disposed an entity of the type Operation goes through the Decide module *If there is delay*. Its content is illustrated in Figure 4.2.18.

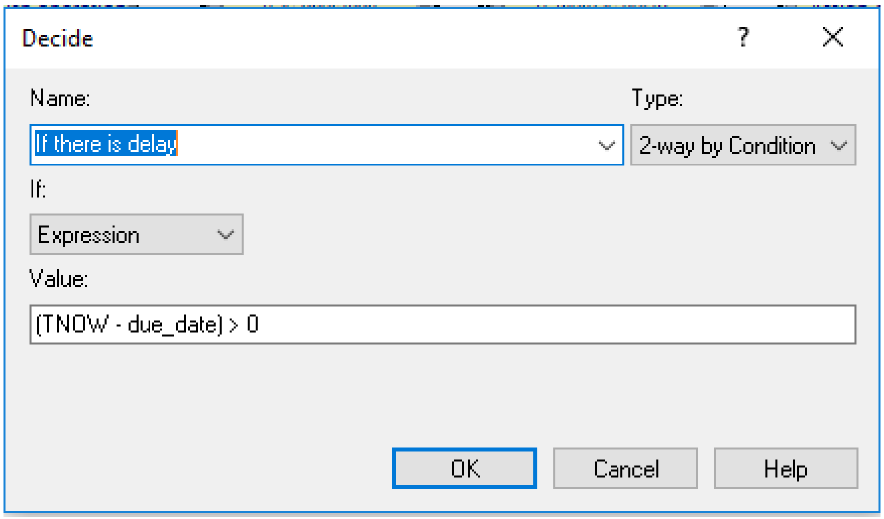

Figure 4.2.18 The Decide module monitoring if there is an operation delay

Arena stores the current simulation time in the system variable TNOW. It shows at what time an operation was actually completed. Therefore a delay can be calculated as the difference between the current time TNOW and the due date of the operation. The operation gets a new attribute *operation_delay,* and its value is equal to that difference if it is greater than 0 and 0 otherwise as it is shown in Figure 4.2.19.

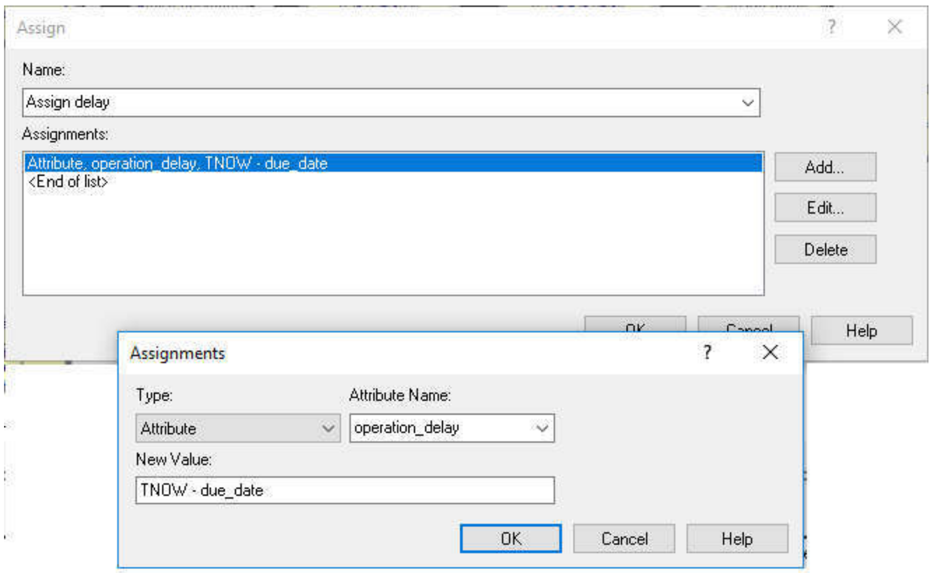

Figure 4.2.19 Assigning a delay to an operation

So as to see the information about operations delays and their completion times in the output of the program it is needed to record these values in the Record module *Record Statistics* before disposing of operations in the Dispose module *Operations don*e. The content of the Record Statistics module is shown in Figure 4.2.20. The Tally Name *delays* here means that the statistics about all operation delays can be found in the report under the string *delays* after the simulation model run.

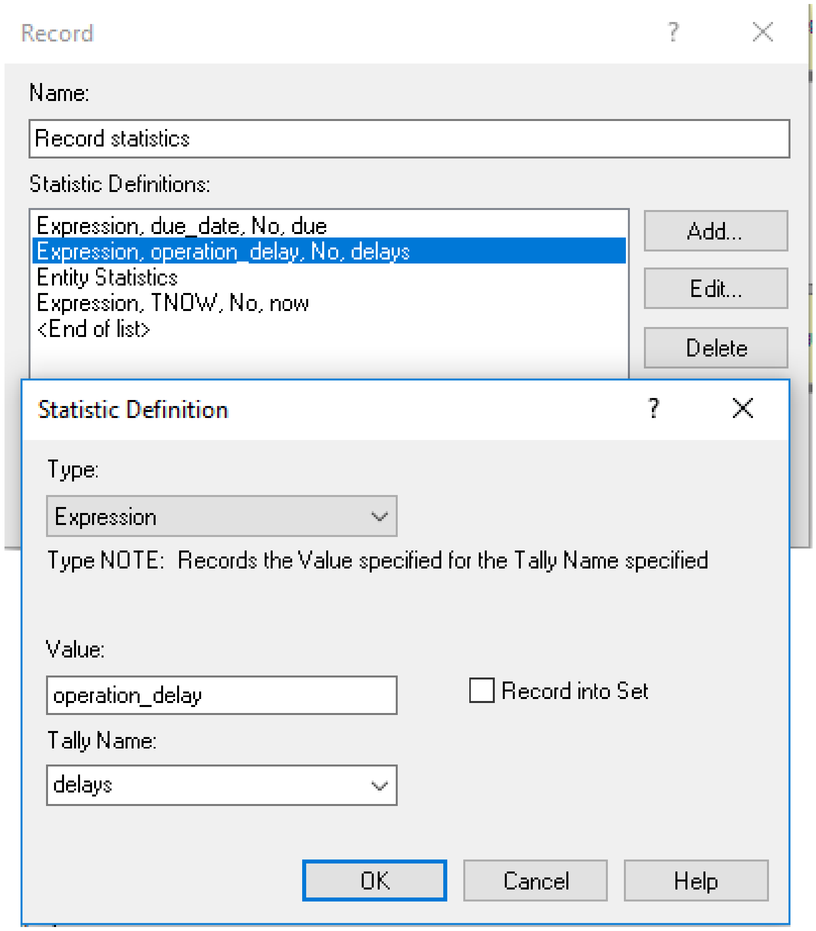

Figure 4.2.20 The Record module to collect information about operation delay and completion time

The full simulation model consisting of four main logic segments described above, is illustrated in Figure 4.2.21.

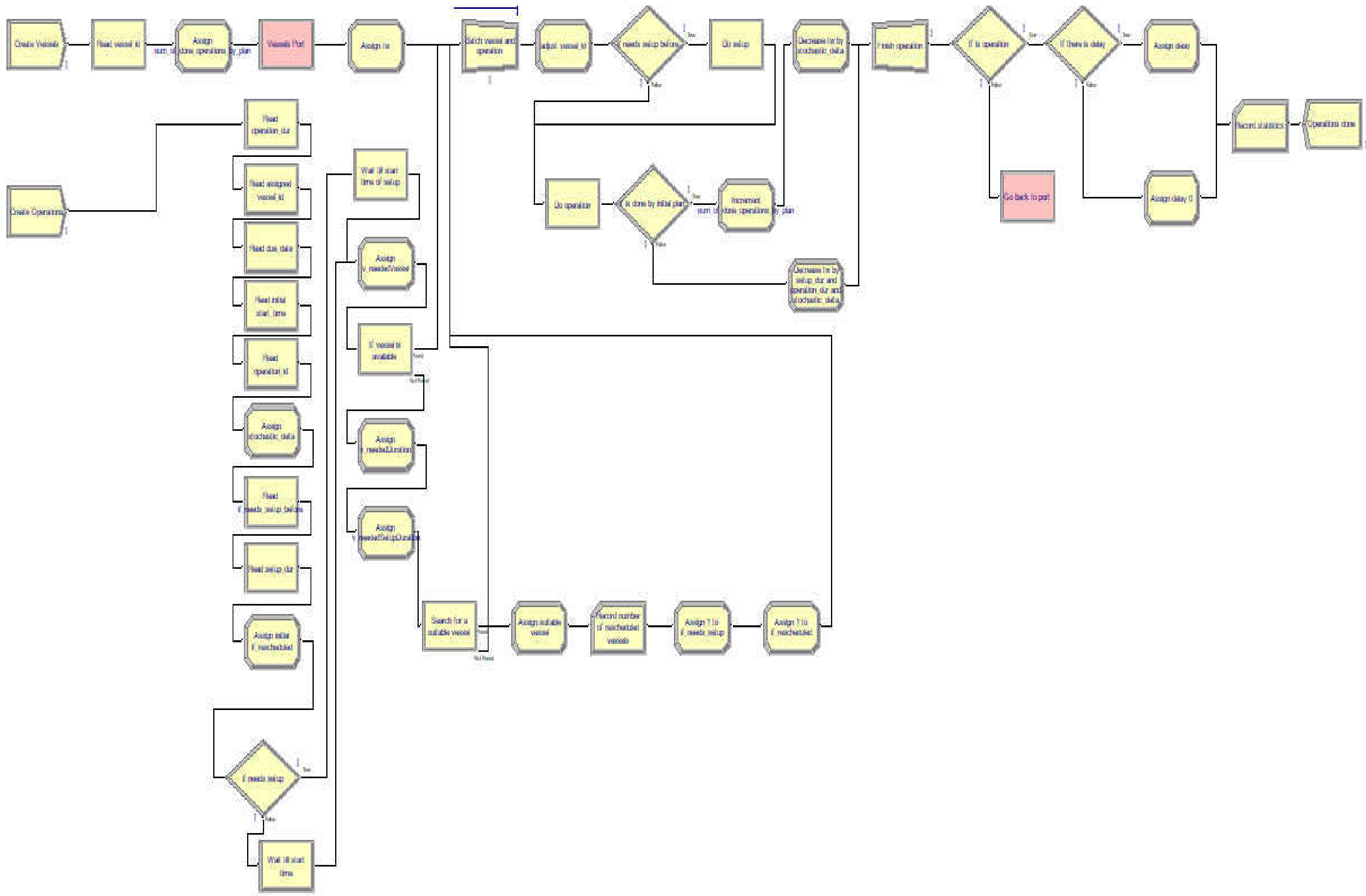

Figure 4.2.21. The full simulation model of implementation of operations by vessel

5 RESULTS

The decision support tool consisting of 2 projects was run on 6 instances containing 25 and 50 operations. All the instance files had their own tightness factor ranging from 1 to 3. The smaller factors characterized tighter windows. Each of the initial schedules had a zero total delay of all operations and was then simulated for 10 replications, and depending on the simulation output results rebuilt for a lower number of vessels and simulated again. This process was repeated several times until significant operations delays occurred. The distribution for random deviations of operations durations was taken TRIA(0, 5, 10) as the best one describing the practical case. The results of the experiments are presented in Table 5.

№ of experiment	*№ of operations*	*Tightness factor*	*№ of performed iterations*	*№ of vessels on the first iteration*	*№ of rescheduled vessels on the last iteration (average value)*	*Suggested № of vessels*
1	25	1	6	10	2	5
2	25	2	8	14	1	7
3	25	3	11	20	3	10
4	50	1	12	26	4	15
5	50	2	21	35	6	15
6	50	3	26	42	5	16

Table 5 Results

For example, the initial schedule for 25 operations with the medium tightness factor generated with the help of constructive heuristic, contained 14 vessels. After simulating this schedule for 10 replications it was discovered that no one of the results had some operation delays and in average 6 operations were assigned different vessels than in the initial schedule as those vessels which were assigned for them initially didn't return to the port in time and couldn't start those operations at the scheduled start time. Having 7 vessels in the schedule still did not show any delays in the simulation and in average 1 operation needed a different vessel so as to be implemented at its scheduled start time. But when the schedule was formed for 6 vessels and simulated the output results showed delays in several simulation replications where the maximum delay was equal to 5 days, and even rescheduling didn't help here to avoid delays as there were no available vessels at time when some operations couldn't be implemented at their initially scheduled start times. Therefore, the conclusion can be made that so as to meet the production plan proposed in the instance file with 25 operations the company should hire at least 7 vessels under the condition that random deviations of processing times of operations vary from 0 to 10 days with an average value equal to 5 days.

The instance file containing 50 operations with the smallest tightness factor required 26 vessels in the generated schedule. After simulation, this schedule for 10 replications presence of delays was encountered, but the average value of delays of all operations was close to 0 days and the maximum delay reached 2 days only. The number of operations which were implemented by different vessels than those which were in the initial schedule on average reached 12. However, having 26 vessels per a long-time contract in practical cases is often impossible for companies, so it was decided to generate the schedule for 50 operations implemented by 15 vessels (what is more real for companies) and to see how many delays will be encountered on average. The generated schedule of voyages for 15 vessels did not have operations delays initially, but after simulating it, the total average delay was discovered to be 3 days with a maximum 14 days. The number of operations which were rescheduled for vessels which were not assigned to them initially was equal to 4 on average.

It is essential that the tighter the time window is between operations release and due dates, the more vessels will be required to process them. Moreover, for some instances, the number of required vessels in the initial schedule was unacceptably big. While simulating it and accepting not significant delays, a more affordable fleet size appeared in results. It should be mentioned that as all companies have different financial resources, they determine an acceptable amount of delays counted in days and they decide whether this amount is acceptable for them in comparison to costs of short-term contracts applied to avoid these delays.

6 CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

In this thesis, the methodology solving Fleet Sizing for Pipe Lay Support Vessels with uncertain operations durations and a decision support tool implementing this methodology was developed. The proposed methodology involves a combination of optimization and simulation methodologies. It suggests schedules of voyages with a minimized number of vessels taking into account the stochastic character of operations implemented by them. This tool is also able to determine a possible total delay of operations which can occur for the specified number of vessels in the schedule so as to give companies an ability to analyze the financial value of risks caused by operations delays and decide whether they are ready to take these risks hiring the chosen fleet size.

The proposed decision support tool was tested on real-based data. And although the results depend on the definite input data much, some results show how much the proposed methodology implying iterative optimization and simulation processes can decrease the fleet size (by approximately 50 %) in comparison to the fleet size suggested by the methodology implying using only the constructive heuristic giving a feasible solution.

There are many ideas on how the proposed methodology can be improved. These improvements imply the extension of *Fleet Sizing for Pipe Lay Support Vessels* problem. This extension relates to having multiple types of not only families of operations but also families of vessels. This will be useful when modern technologies and researches show the world several types of PLSVs implementing certain types of operations. The extension of the problem also relates to stochastic setup durations, while in our work we considered them to have a deterministic character. There are many cases when stochastic factors can disrupt a production plan during not only operations implementation but also loading of equipment and navigation to wells.

The idea of improvement of the proposed methodology also lies in using or developing an algorithm giving a more optimal schedule of vessels to be simulated, i.e. replacing the proposed heuristic based on the ATC algorithm and adjusted to PLSV Scheduling Problem with another one giving more optimal schedules. Also, the more precise distribution of deviations of operations durations can be defined with the help of historical data and then used during the simulation phase of the methodology.

The improvements related to the proposed decision support tool imply the creation of a more user-friendly interface with an easier way of run. Because in our thesis, the tool responsible for generating schedules is a console application while analysts in companies usually like to work with desktop applications with more understandable conditions of generating results.

REFERENCES

Arora, S., Lund, C., 1996. Hardness of approximations. Approximation algorithms for NP-hard problems, PWS, Boston, 399-446.

Baker, K. R., Bertrand, J. W. M., 1981. An investigation of due-date assignment rules with constrained tightness. Journal of Operations Management 1(3), 109-120.

Biskup, D., Herrmann, J., Gupta, J.N.D., 2008. Scheduling identical parallel machines to minimize total tardiness. International Journal of Production Economics 115, 134-142.

Coffman, E., Garey, M., Johnson, D., 1978. An application of bin packing to multi-processor scheduling. SIAM J. Comput.. 7(1), 1-17.

Cristiansen, M., Fagerholt, K., Ronen D., 2004. Ship routing and scheduling: Status and perspectives. Transportation Science 38, 1-18.

Cunha, V., Santos, I., Pessoa, P., Hamacher, S., 2018. An ILS heuristic for the ship scheduling problem: application in the oil industry. International Transactions in Operational Research 00, 1-22.

Fang, K., Wang, Y., 1993. Number-theoretic methods in statistics. Monographs on Statistics and Applied Probability 51, 104-155.

Graham, R. L., 1966. Bounds for Certain Multiprocessing Anomalies. Bell System Technical Journal 45, 1563-1581.

Lee, Y. H., Pinedo, M., 1997. Scheduling jobs on parallel machines with sequence dependent setup times. European Journal of Operational Research 100, 464–474.

Lenstra, J. K., Rinnooy, K. A. H. G., 1979. Computational complexity of discrete optimization problems. Annals of Discrete Mathematics 4, 121-140.

Maisiuk, Y., Gribkovskaia, I., 2014. Fleet sizing for offshore supply vessels with stochastic sailing and service times. Procedia Computer Science 31, 939–948.

Mokotoff, E., 2001. Parallel machine scheduling problems: a survey. Asia-Pacific Journal of Operational Research 18, 2, 193–242.

Pfund, M., Fowler, J., Gadkari, A., Chen, Y., 2008. Scheduling jobs on parallel machines with setup times and ready times. Computers and Industrial Engineering 54(4), 764-782.

Queiroz, M. M., Mendes, A. B., 2011. Heuristic approach for solving a pipe layer fleet scheduling problem. Sustainable Maritime Transportation and Exploitation of Sea Resources 1, 1073–1080.

Shyshou, A, Gribkovskaia, I, Barcelo J., 2010. A simulation study of the fleet sizing problem arising in offshore anchor handling operations. European Journal of Operational Research 203(1), 230-240.

Su, H., Pinedo, M., Wan, G., 2017. Parallel machine scheduling with eligibility constraints: A composite dispatching rule to minimize total weighted tardiness. Naval Research Logistics 64, 249-267.

Vepsalainen, A., Morton, T., 1987. Priority Rules for Job Shops with Weighted Tardiness Costs. Management Science 33, 1035-1047.