

FAMAS - NewCon

A generator program for stacking in the reference case



Report Econometric Institute
EI 9943/A

Contents

1. Introduction.....	3
2. Objectives of the simulation models	5
2.1 Various models	5
2.2 Experimental design.....	6
3. Modelling the arrivals and departures of ships	7
3.1 Scenario 'High'	7
3.1.1 Arrivals of jumbo and deepsea ships.....	8
3.1.2 Truck arrivals.....	9
3.1.3 Arrivals of shortsea/feeder, rail and barge	9
3.1.4 Determination of flows	12
3.2 Scenario 'Low'	13
4. Description of the long-term stacking simulation program.....	15
4.1 Program description	15
4.1.1 Ship and container generator program	15
4.1.2. Test program.....	20
4.1.3 Execution of stacking algorithms	21
5. Results of the test program	23
5.1 The generated number of containers.....	23
5.2 Arrivals of trucks and the other modalities	24
5.3 The time a container spends in the stack.....	26
6. References.....	28
Appendix I - Categories.....	29
Appendix II - Input variables.....	31
Appendix III – Flows (input).....	32
Scenario ‘Low’	32
Scenario ‘High’	33
Appendix IV – List of abbreviations and definitions	36

1. Introduction

This report originates from the FAMAS-NewCon research project. FAMAS (First-All-Modes-All-Sizes) is a comprehensive research programme initiated by the Centre for Transport Technology (CTT) and aims at the development of a new generation of container terminals with a guaranteed service level for all modalities. It builds on and tries to improve the robot technology developed for ECT's (Europe Combined Terminals) Delta/Sea-Land (DSL) and Delta Dedicated East (DDE) terminals. This means that both Automatic Stacking Cranes (ASCs) and Automated Guided Vehicles (AGVs) will be considered in the plans. The Quay Cranes, however, will be manned.

The FAMAS project has a number of subprojects of which NewCon is one. NewCon considers terminal control, transport, administration and planning. A key challenge within NewCon is to handle a Jumbo container ship of 8000 TEU within 24 hours in which some 6000 containers are unloaded and loaded.

The NewCon project concentrates on the conceptual design of the whole terminal. The starting points of the NewCon project have been captured in a definition study (see Celen et al. (1997)). The design of a quay crane will be done in another subproject; accordingly the quay crane and its performance is a given for NewCon. Instead, NewCon concentrates on the feeding process of the quay crane and hence on the stack and the quay transport. The intention is that the feeding process guarantee a 100% feeding of the quay cranes and that in the loading process there will always be a loaded AGV present at the crane when needed. Technical specifications of AGVs, however, are a subject of another project, viz. Smagic. Hence NewCon focuses on

- (1) the layout and sizing of the stack (nr of lanes, length, height of stack)
- (2) the planning and control concepts
- (3) AGV routing

Four research cases have been defined within NewCon. The first one is a reference case with a circular routing of AGVs, like in the present Delta/Sea-Land Terminal. The second case considers a cross routing of AGVs, i.e. AGVs cross the quay perpendicular to the stack, directly to the quay crane. The third case considers an Automatic Lift Vehicle and the last case concerns a high quay, where a substack decouples the feeding process from the stack and the quay cranes. The idea of the reference case is to construct a situation comparable to an actual situation, which serves as a validation of the models to be developed. Both the Delta/Sea-Land layout and an upscaled layout will be considered. The main idea in the cross case is to minimize conflicting AGV behaviour. Finally, both in ALV case as in the high case there is a decoupling of the queue before the quay crane and the export process from the stack. This report considers the reference case, which has been studied in the first phase of the research. Many elements are however also suited for the other cases.

The NewCon project team consists of teams from the TRAIL Research school supplemented with participation from ECT (ir. R. van de Ham and ir. A. Nagel) and with ir. F. Nooijen as project leader. The two TRAIL teams originate from the TU Delft (the section of Evers) and the Erasmus Universiteit Rotterdam (section of Dekker). The TUD team will study the quay layout and the AGV routing, while the EUR team will deal with stacking. The idea is to develop three separate simulation programs that will be combined later on. Initially the plans were that the EUR would develop two

simulation models, one for short-term stacking and one for long-term stacking. The development on the short-term stacking however, was later frozen in the project. The TUD team will develop a simulation program for the AGV routing which will later be extended with short-term stacking. Short-term stacking involves the scheduling of the stacking operations for the loading and unloading of one (jumbo) ship. It reveals the bottlenecks in the import and export capacities of the stack. The long-term stacking deals with a much longer period and concerns the total stack storage capacity.

This report will deal only with long-term stacking. It gives a description of the generator model as well as its implementation in a simulation program. In the generator model arrivals and departures of containers are modelled over a long period (in the implementation mainly 12 weeks). The program yields as output files with this information on an individual container level. Next to that a stacking model (and program) is built which will use this information to put the containers at the prescribed time at a to-be-determined place in the stack as to export them at another, prescribed time. The advantage of this separation is that different stacking configurations and strategies can be compared using the same input file. The program described in this report dates from August 1999, which is of importance with possible future changes in the model.

The structure of this report is as follows. In chapter 2 we describe the different simulation models dealing with stacking. Chapter 3 starts with the requirements and main ideas behind the long-term stacking program. It gives an analysis and modelling of the cross-streams as defined in the origin-destination matrix. Chapter 4 then describes the actual simulation program, by explaining its objects, its input and output. Chapter 5 then gives the results of a test program, which is used to see whether the specifications are met by the program. Chapter 6 discusses the design of experiments, whereas their results are discussed in chapter 7. The conclusions and some recommendations for future research are presented in chapter 8.

2. Objectives of the simulation models

2.1 Various models

Increasing the production of a robotised container terminal is a difficult problem in which many factors play a complex role. Just upscaling an existing terminal is not a good idea. A proper tuning between the various subsystems is of great importance. To answer the many questions it has been decided to split up the problem in pieces, viz.

- short term physical quay transport
- short term stacking operations
- long term stacking operations

The motivation behind this choice will be explained hereafter.

One of the present main problems in the quay transport is that the AGVs have many physical conflicts during their movements. These traffic problems cause for delays in their travelling to and from the quay cranes. As the loading sequence of the containers is strictly fixed at the moment, a late arrival of an AGV at a quay crane (QC) may imply that the latter has to wait, with a loss of production as a result. The first line of research focuses on these problems and applies a simulation model in which the physical movements of the AGVs are modelled in detail. One tries various layouts, routes and travel control scripts to handle the traffic problems. Initially, the stack is modelled in a simple way. The stacking equipment (like Automatic Stacking Cranes) are not modelled; one decided to work with a stack response function, indicating the average delay the stack has to deliver a container. In this model no ASC control is modelled.

Control of the stacking equipment is a central issue in the second model. The idea is that short-term overload of the stacking cranes is one of the main bottlenecks in the container feeding process to the quay cranes. The reason is that the cycle time of a stacking crane, i.e. the time needed to retrieve a container from the stack and put it on a ASC (can be some two minutes), is larger than the cycle time of the quay crane (less than a minute). This implies that many more stacking cranes are needed to feed the quay cranes. In the present DSL terminal there are some 27 stacking cranes to feed some four to six quay cranes. As it is very difficult to exchange containers between stacking lanes on the short term, any above average demand for containers out of a single stacking lane will cause capacity problems. Where reshuffles and landside demand for containers complicate the capacity problems, buffering at the quay cranes helps. Yet it is not easy to increase the buffer of AGVs at the quay cranes as that would take up much space. An interesting idea is therefor to do some work in advance, like pre-positioning of containers and carrying out the necessary reshuffles. This does require however, a more sophisticated ASC planning and scheduling.

All these considerations led to the plan to develop a second simulation model in which both AGVs and ASCs are modelled explicitly. To keep the model simple and tractable it was decided to model the travelling of the AGVs not explicitly, as that was already taken into account in the first model. The idea is to simulate the traffic congestion by modelling the travelling delays by statistical distributions whose parameters would be determined from the first model. In this way, one could better concentrate on the planning and scheduling of the AGVs and ASCs. The stack would,

however, be modelled in detail in the second model, with detailed information about the position of the containers to be loaded in the stack. An important aspect in this respect is the distribution of the containers to be loaded in a jumbo container vessel. This distribution however, can only be determined by following import of all containers and applying the stacking rules in detail. This would however lead to the simulation of the operations over a long period, implying a lengthy simulation. To prevent such it was decided to deal with the long-term stacking in a third model and simulate a short period only in the second – short-term stacking - model (the loading and unloading of one ship). The third model would then yield the initial stack distribution to be used in the second model. This separation would also allow a separate study of short term planning and scheduling of equipment and of the long-term stacking rules. The simulation models would then be simpler and the work can be executed in parallel.

The short-term stacking model is described in Meersmans et al. (1999). This report focuses on the long-term stacking model.

2.2 Experimental design

The main actions in a stacking model are the arrival, stacking and departure of containers over a long period. An important aspect in this respect is the information available upon arrival of a container about the way the container will leave the stack. This concerns the modality and where appropriate the ship.

Several criteria can be used in stacking: the number of reshuffles and the positioning of the containers with respect to their export modality. In case of import containers the positioning is less important since there is a more even pattern of collection. A reshuffle can be caused in two ways. First of all the container can be put below other containers from the same ship which have to be loaded later. The large call size of a jumbo ship makes this aspect more important. Secondly, the container can be below containers of other ships or modalities that have to be collected later.

Large ships and jumbo container ships especially create a high demand on the container stacking capacity: a large number of containers have to be put into the stack and retrieved from the same stack in a short amount of time. Moreover, the moment at which all containers have been unloaded from a jumbo ship is the moment when the stack storage capacity is used maximally. The presence of container ships thus creates the peaks in capacity usage. Moreover, it is also very important for the stack handling to know where a container is offered to the stack: from land or sea side. Accordingly both the storage into and retrieval out of the stack have to be modelled precisely. This modelling should then occur under the requirements of the origin-destination matrix as specified in the NewCon definition study (see Celen et al. (1997)). This means that arrivals and number of unloaded and loaded containers have to be modelled in detail. This will be done in the next chapter. Both the low (about 600.000 containers annually) and the high scenario (3.100.000 containers annually) will be considered.

3. Modelling the arrivals and departures of ships

In this section we will explain how we modelled the flows between the different transport objects. Two possible scenarios are considered. The first scenario handles about 3.1 million containers, whereas the second scenario has a throughput of approximately 620,000 containers per year. These scenarios are called the scenario 'High' and the scenario 'Low'. The idea is to come up with a fixed schedule for the arrival of deepsea and jumbo container ships. Due to the large number of short-sea and feeder vessels with bringing or taking each a low number of containers, we decided not to model each feeder or shortsea ship in detail, but to model only their total flow. The same was done for the truck, rail and barge modalities. We did allow for variations in these flows.

3.1 Scenario 'High'

The following destination matrix with 3.1 million container visits is derived from Celen et al. (1997), with the only difference that we have taken the shortsea and feeder modes together:

From/to	<i>Jumbo</i>	<i>Deepsea</i>	<i>Shorts./feed.</i>	<i>Truck</i>	<i>Rail</i>	<i>Barge</i>	Total
<i>Jumbo</i>	0	200000	325000	35000	80000	110000	750000
<i>Deepsea</i>	200000	100000	100000	20000	30000	50000	500000
<i>Shorts./feed.</i>	325000	100000	0	95000	185000	270000	975000
<i>Truck</i>	35000	20000	95000	0	0	0	150000
<i>Rail</i>	80000	30000	185000	0	0	0	295000
<i>Barge</i>	110000	50000	270000	0	0	0	430000
Total	750000	500000	975000	150000	295000	430000	3100000

The same report states that the average number of moves per jumbo equals 6000 and that the average number of moves per deepsea ship equals 3000. Based on this information, the number of jumbo and deepsea calls would be:

- Jumbo : $(2 * 750000) / 6000 = 250$ calls per year
- Deepsea : $(2 * 500000) / 3000 = 333$ calls per year (rounded)

Based on a year with 50 weeks this would result in 5 jumbo calls and 6,67 deepsea calls per week. In order to maintain this ratio and to obtain an integer number of calls per period the period length is set to three weeks. During this three weeks there are 15 jumbo arrivals and 20 deepsea arrivals. This results in the following flows per period for a period with length three weeks:

From/to	<i>Jumbo</i>	<i>Deepsea</i>	<i>Shorts./feed.</i>	<i>Truck</i>	<i>Rail</i>	<i>Barge</i>	Total
<i>Jumbo</i>	0	12000	19500	2100	4800	6600	45000
<i>Deepsea</i>	12000	6000	6000	1200	1800	3000	30000
<i>Shorts./feed.</i>	19500	6000	0	5700	11100	16200	58500
<i>Truck</i>	2100	1200	5700	0	0	0	9000
<i>Rail</i>	4800	1800	11100	0	0	0	17700
<i>Barge</i>	6600	3000	16200	0	0	0	25800
Total	45000	30000	58500	9000	17700	25800	186000

Later on, it appeared that these flows are not compatible with other input data. The norm ships for jumbo and deepsea (see Appendix II) do not exactly load and unload 3000, 1500 containers respectively. Besides that, the number of containers to be loaded and to be unloaded is neither the same for the jumbo or the deepsea ships. The number of containers to be unloaded equals 2913 for

a jumbo and the number of containers to be loaded equals 3190. For deepsea ships these numbers are 1682 and 1843 respectively. A direct consequence of this specification is that the flow matrix can never be symmetric, when using the norm ships. For both the jumbos and the deepseas the number of containers to be unloaded is less than the number of containers to be loaded.

Adjusting the former OD matrix with the correct numbers of containers to be (un)loaded and under the restriction that there are still 186000 container visits per three-week period yields the following flows during one period:

From/to	<i>Jumbo</i>	<i>Deepsea</i>	<i>Shorts./feed.</i>	<i>Truck</i>	<i>Rail</i>	<i>Barge</i>	Total
<i>Jumbo</i>	0	11660	18150	2035	4825	7025	43695
<i>Deepsea</i>	13455	6720	7330	1350	1945	2840	33640
<i>Shorts./feed.</i>	19350	10000	0	4835	9380	13675	57240
<i>Truck</i>	2190	1840	4835	0	0	0	8865
<i>Rail</i>	5235	2700	9385	0	0	0	17320
<i>Barge</i>	7620	3940	13680	0	0	0	25240
Total	47850	36860	53380	8220	16150	23540	186000

The flows from all modes to the mode jumbo are all divisible by 15, the flows from all modes to the mode deepsea are all divisible by 20. The reason for this will become clear below.

Now that we know the flows between the different modes, we have to determine the flows between the objects. We not only want to know the flow from the deepsea modality to the jumbo modality but also the flow from, e.g. the third deepsea in a period to the tenth jumbo in the same period.

In determining these flows we should, apart from the modal split, also consider an average dwell time of 3.9 days per container (see Celen et al. (1997)). That's why it is important to know the arrival times of the objects.

3.1.1 Arrivals of jumbo and deepsea ships

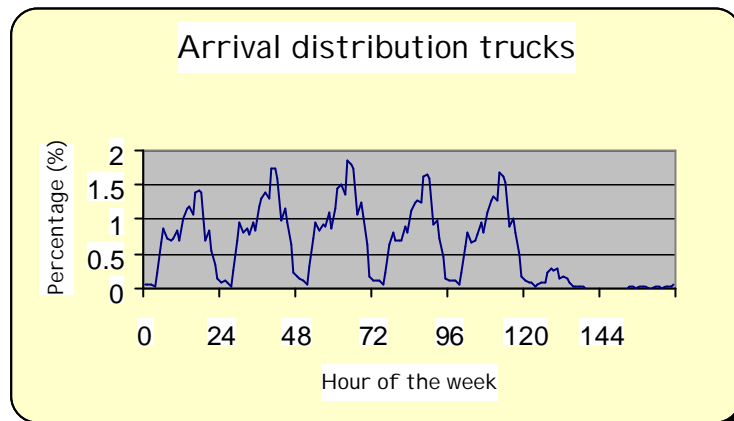
For the jumbo and deepsea ships we have chosen for an 'equidistant' schedule. This means that all ships of the same mode have the same scheduled inter-arrival time with the previous or next ship. Next, disturbances on this schedule are introduced in the simulation. The arrivals are disturbed with a random drawing out of the uniform distribution on the interval from -0.5 days to 0.5 days. According to the fixed scheme jumbos arrive with inter-arrival times equal to 33.6 hours ($3 \text{ weeks} \times 7 \text{ days} \times 24 \text{ hours}$ divided by 15 jumbos). Deepseas arrive with inter-arrival times equal to 25.2 hours ($3 \text{ weeks} \times 7 \text{ days} \times 24 \text{ hours}$ divided by 20 deepsea ships). If we define Monday 0:00, week 1 to be the arrival time of the first jumbo as well as the first deepsea in the period, then the successive arrivals are shown below:

Jumbo: week 1 – Mon 0:00, Tue 9:36, Wed 19:12, Fri 4:48 and Sat 14:24
 week 2 – the same
 week 3 – the same

Deepsea: week 1 – Mon 0:00, Tue 1:12, Wed 2:24, Thu 3:36, Fri 4:48, Sat 6:00 and Sun 7:12
 week 2 – Mon 8:24, Tue 9:36, Wed 10:48, Thu 12:00, Fri 13:12, Sat 14:24 and Sun 15:36
 week 3 – Mon 16:48, Tue 18:00, Wed 19:12, Thu 20:24, Fri 21:36 and Sat 22:48

3.1.2 Truck arrivals

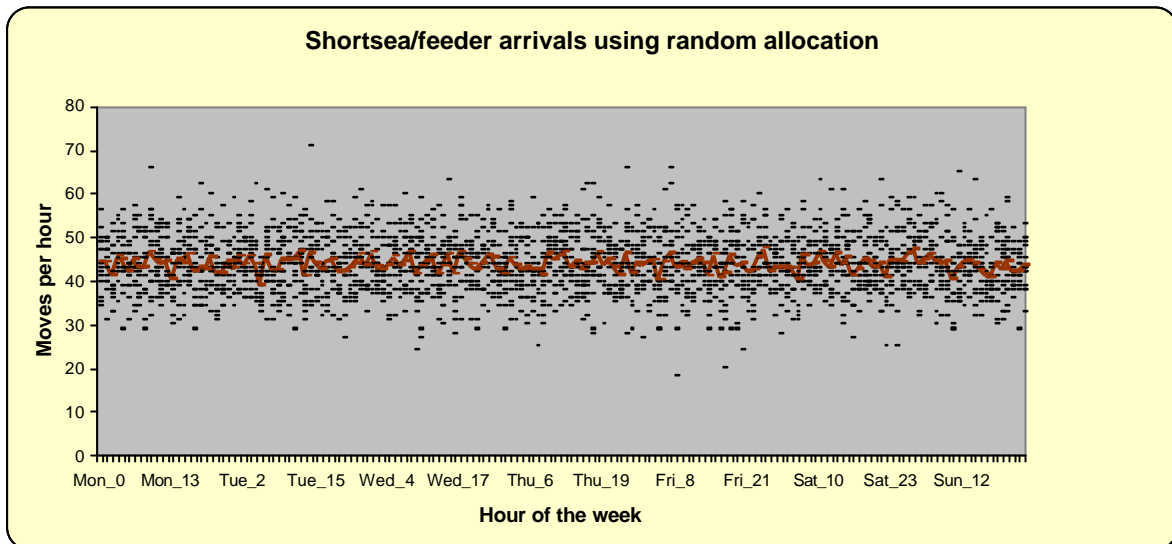
The other modes are shortsea/feeder, truck, rail and barge. Containers for these modes are treated as individuals. The arrivals of the trucks are known to show a pattern over time. We will use a weekly pattern, shown in the graph below. This pattern has been deduced from data about the arrivals at the present terminals (data were supplied by ECT).



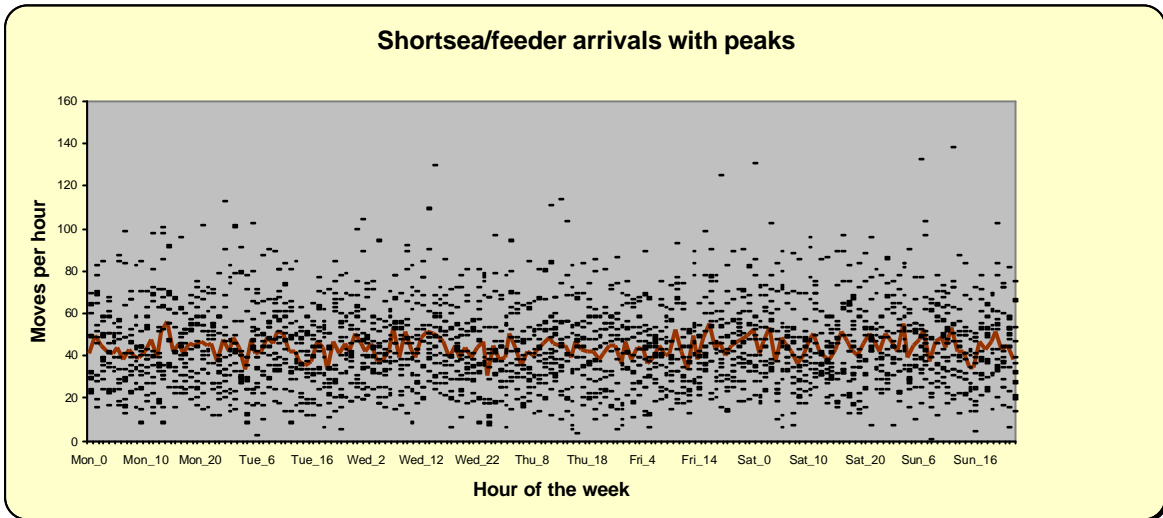
In this graph the hour 0 represents Monday 0:00 until Monday 1:00, the hour 1 means Monday 1:00 till Monday 2:00, and so on. The last hour is 167 meaning Sunday 23:00 till Monday 0:00. Notice the daily pattern and the fact that there are almost no arrivals during the weekends.

3.1.3 Arrivals of shortsea/feeder, rail and barge

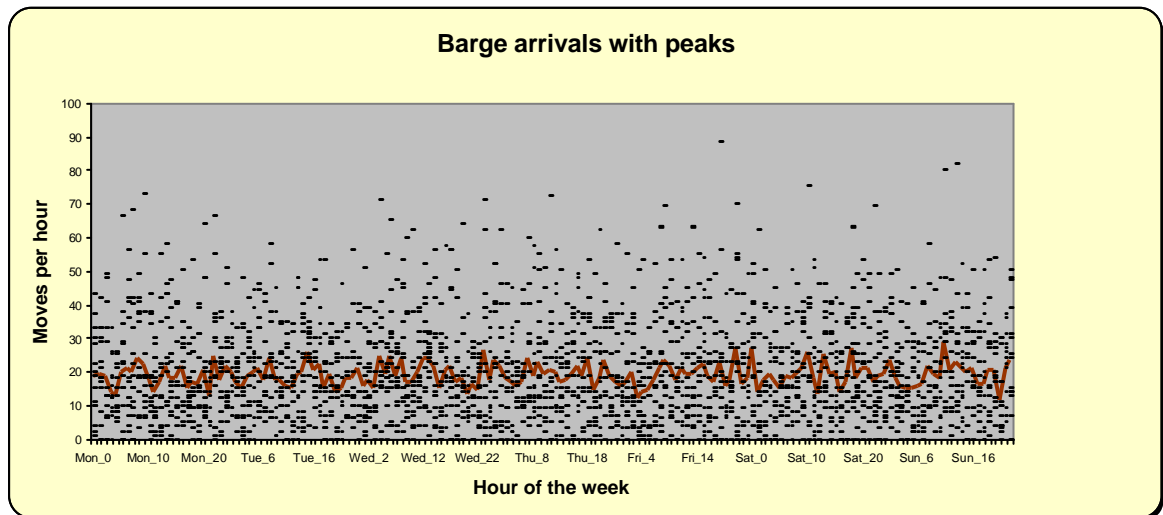
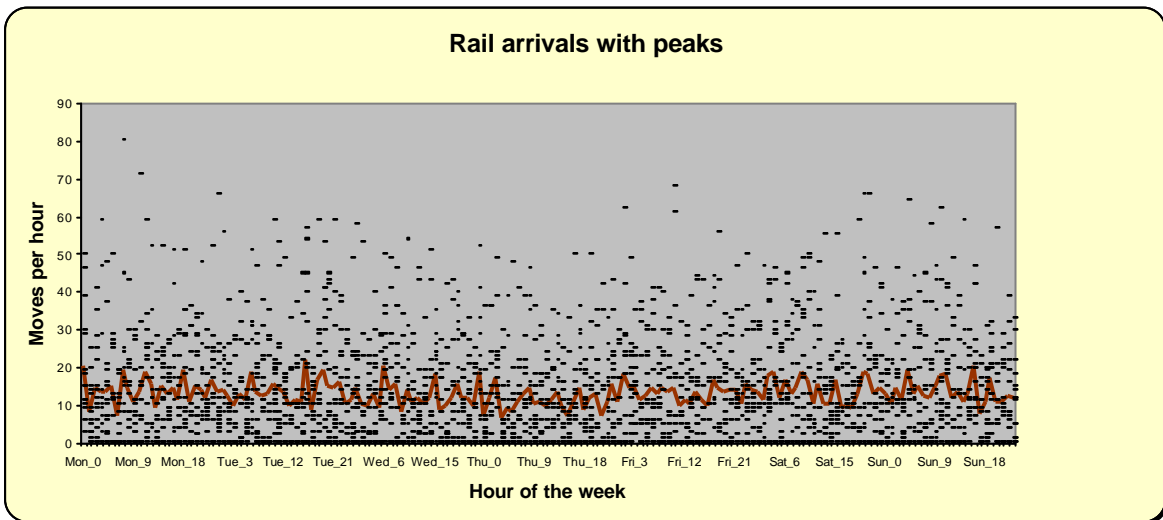
In contradistinction to the arrivals of trucks, no significant pattern was found for the modes shortsea/feeder, rail and barge. That's why we decided to randomly dedicate one of the possible 168 hours during one week to each container move for these modes. This means that for all such moves one hour is chosen with all of the 168 hours having the same possibility to be chosen. After that randomly a time (in seconds) within that hour is generated. Later on however, a remark was made concerning those arrivals. Although, in the long run, there was no significant difference in the number of moves for these modes in the hours of a week, the existence of peaks in the arrivals couldn't be neglected. Using this algorithm could be an oversimplification of truth, because of the small variation in values. As shown in the following graph, this way of assigning hours to all moves gives a pattern with almost all observations close to the average number.



This graph is based on fifteen weeks. So there are fifteen observations per hour of the week plus the average value of those observations. Because of the lack of peaks, another way of assigning hours to the moves has been determined. This method uses the data from the Excel file *Data_M~1.xls* (ECT, 20-07-'99). These are data concerning the current terminals over the weeks two up to and including sixteen of 1999. This results in 2520 (15 times 168) observations. Those observations are all taken together, resulting in a distribution for the number of arrivals during one hour of the week. Then for every week, for every hour in that week, a number of moves during that hour is drawn according to this distribution. The total number of moves during one week is however fixed. The number obtained by sampling will probably not be equal to this fixed value (especially not in the case of shortsea/feeder moves, where this fixed value is well over thirteen times the average number obtained). Therefore we constructed a method to scale the number of moves up (or down) to this fixed value. As long as the fixed value isn't reached, one of the 168 hours is chosen at random (sampling without replacement). Then a number of moves between one and ten is chosen, also randomly. This number is added to (subtracted from) the number of moves for that hour. Of course, in the case of too many moves, no more moves can be subtracted than the number of moves dedicated to that hour. Furthermore, the algorithm ends whenever the total number of moves generated equals the desired number of moves during that week (Note that the number of moves to be added (removed) is also restricted by the desired number. No more moves than the difference can be added or removed). This results in the following graph.



Clearly, in this case, the variance is much higher. The algorithms for rail and barge moves are equal to the method described here. Only difference is that the number of moves and the distribution of moves per hour differ from the ones above. The graphs for these modes are shown below.



It is important to understand that these graphs are generated using the random function. This means that the numbers actually used in the program will most certainly not be equal to the numbers above. They will however show the same kind of patterns as shown above.

3.1.4 Determination of flows

Using the input mentioned above, one can determine the flows between the objects. For the containers that are loaded with a jumbo or a deepsea, we can determine which modes brought these containers. This comes down to dividing the numbers in the previous table with flows in the columns jumbo and deepsea by 15 and 20. This is the reason why this numbers are exactly dividable by 15 or 20. In this way, the flows are the same for every jumbo or every deepsea. This offers the next table:

From/to	<i>Jumbo</i>	<i>Deepsea</i>
<i>Jumbo</i>	0	583
<i>Deepsea</i>	897	336
<i>Shorts./feed.</i>	1290	500
<i>Truck</i>	146	92
<i>Rail</i>	349	135
<i>Barge</i>	508	197
Total	3190	1843

First of all we determine the flows between jumbos and deepseas. Every jumbo gets 897 containers from the five deepseas coming closest before this jumbo according to the fixed scheme. To avoid “late comers” (containers that should leave the stack before they arrive in it) the next restriction is added. A deepsea bringing containers for this jumbo should, according to the fixed scheme, arrive at least 48 hours before the jumbo. The disturbances included, this results in a minimum time between arrivals of the deepsea and the jumbo of 24 hours, which is the processing time of a deepsea (and of a jumbo). In this manner a container to be loaded into the jumbo is always present in the stack at the loading time.

Every deepsea receives 583 containers from the four jumbos coming closest before this deepsea according to the fixed scheme. Again, the restriction above has to be taken into account. So the jumbos have to arrive at least 48 hours before the deepsea arrives.

Furthermore, every deepsea receives 336 containers from the six deepsea ships coming closest before this deepsea, however with the restriction there are at least 48 hours between the arrivals according to the fixed scheme. Because of the deepseas having inter-arrival times equal to 25.2 hours, the only deepsea to be too late is the deepsea coming directly before this deepsea. The deepsea ship before that always comes in time (, because 2 times 25.2 is 50.4, which is more than 48).

As appears from the table above, jumbo and deepsea ships also receive containers from the other modes. Hoping (and expecting) that the average time in the stack for these containers will be approximately 3.9 days, the other modes aren’t divided over the days of the week. Only in the simulation the arrival time of the container brought by such a mode is determined. A moment in time during a week is drawn (for trucks according to the distribution above, for the other modes completely arbitrary). Whenever this moment comes before the ship arrival in a week (based on the sequence Monday, ..., Sunday) the arrival time (the moment in time where the container arrives in the

stack) is correct. If not, a week is subtracted from this time. So, the container arrives a week earlier at exactly the same day and the same time. We expect that, using this method, these containers spend an average time of half a week, thus 3.5 days in the stack. This could differ through the way the arrival times of the jumbos and deepseas are chosen. Should it happen that the average time a containers spends in the stack differs too much, then the time a container enters the stack can be restricted by adding a minimal time-interval between arriving in the stack and leaving the stack (whenever the average time in the stack is too short) or by adding a maximum constraint for the time between entering and leaving the stack (whenever the time in the stack is too long). Whatever is the case will come forward after executing the test program.

Of course, there are also flows from jumbo and deepsea ships to the other modes, or in other words containers that are brought by jumbos or deepseas and that are taken away by one of the other modes. We already know how many of the containers, brought by jumbo or deepsea, are taken away by jumbo or by deepsea. Besides we also know how many containers every jumbo or deepsea unloads. The containers that aren't assigned to any mode are divided into the same proportion as the containers to be loaded by these modes. The assignment of a container to an object takes, again, only place during the simulation. This time an object arrives in the same week as the jumbo or deepsea whenever the arrival time of the object comes after the arrival time of the jumbo/deepsea. If not, the arrival of the other mode is exactly one week later.

Last of all, there are the flows between the other modes mutually. These flows are only given as flows between the modes, not as flows between the objects. During the simulation, an entering-time and a leaving-time are determined (according to the distribution concerned). The entering-time and the leaving-time are in the same week when the leaving-time comes after the entering-time. If not, the leaving time comes a week later.

All calculated flows can be found in appendix III.

3.2 Scenario 'Low'

For this scenario, the flows between the modalities are all one-fifth of the flows for the scenario 'High'. Therefore the number of jumbos per three-week period is three, and the number of deepsea ships is four. The flows per period are given in the table below.

From/to	<i>Jumbo</i>	<i>Deepsea</i>	<i>Shorts./feed.</i>	<i>Truck</i>	<i>Rail</i>	<i>Barge</i>	Total
<i>Jumbo</i>	0	2332	3630	407	965	1405	8739
<i>Deepsea</i>	2691	1344	1466	270	389	568	6728
<i>Shorts./feed.</i>	3870	2000	0	967	1876	2735	11448
<i>Truck</i>	438	368	967	0	0	0	1773
<i>Rail</i>	1047	540	1877	0	0	0	3464
<i>Barge</i>	1524	788	2736	0	0	0	5048
Total	9570	7372	10676	1644	3230	4708	37200

We're using the following scheme of arrivals according to the fixed scheme:

- Jumbo : Thu 0:00 wk 1, Wed 0:00 wk 2 and Tue 0:00 wk 3
- Deepsea : Mon 0:00 wk 1, Sun 0:00 wk 1, Sat 0:00 wk 2 and Fri 0:00 wk 3

In this manner, the inter-arrival times of two ships equal three days.

Furthermore, all ships get the same amount of containers from all modes as for the scenario 'High'. Because of this the flows from the other modalities to a jumbo or deepsea ship are identical to those determined in the previous section.

For the flows between the jumbos and deepseas there's the next remark to be made. Because of the inter-arrival time between two subsequent ships being as high as three days, the ship directly before this ship brings mostly all of the containers taken away by a ship. The only exception to this rule is that jumbo 1 receives containers from deepsea 1 as well as deepsea 4 (in the previous period) and that jumbo 3 brings containers for deepsea 4 as well as for deepsea 1 (in the next period). This is caused by the number of deepseas being one higher than the number of jumbos per period.

When the flows between the jumbos and deepseas are determined it is possible to divide the containers coming from one of these modalities and leaving with one of the other modalities according to (approximately) the same ratios as mentioned in the previous section. Finally, the flows between the other modalities are exactly one-fifth of the flows for the scenario 'High'.

The calculated flows can again be found in appendix III.

4. Description of the long-term stacking simulation program

For the simulation, we have chosen to divide the program into three separate programs. The first program generates the ship arrivals and the container visits for a specified period. A second program is used to check for possible errors in the first program. A third program uses the generated ships and container moves to execute a certain stacking algorithm for a certain stack configuration. By running this program a number of times for different stacking algorithms and different stack dimensions a proper comparison is made.

Initially the plan was to put the programs into one big program. The reason for not doing this is threefold. First of all, it is clear to see that a few small programs are better understandable than one big program. Besides, having to generate containers for a pretty long time period again and again is a rather time-consuming business. Thirdly, programming in this way gives the extra advantage that the ships and containers are exactly identical for all tested stacking algorithms.

4.1 Program description

In this section we will separately describe the three programs mentioned above.

4.1.1 Ship and container generator program

This program is the most complicated program of the three programs written. We will discuss the program in a few steps. First of all, we will describe the input needed to execute the program. After that, the procedures used in the program are explained, as is their use in the program by means of a piece of pseudo-code. Finally, further information about the output generated by the program will be given. But first of all we will describe the necessary attributes of a container we want to use in the programs:

- ID-number. This is a unique number. The first container gets number 1, the second number 2, and so on.
- Category (only for containers to be loaded by jumbo or deepsea). This number indicates the size, the weight and the destination port of a container. The categories can be found in the enclosures.
- Unload-object. This is the object (specific ship or other modality) that brings the container to the terminal.
- Load-object. This is the object (specific ship or other modality) that takes the container away from the terminal.
- Unload-time. The time the container is unloaded.
- Load-time. The time the container is loaded.
- The position in the ship that unloads the container (bay, horizontal and vertical position).
- The position in the ship that loads the container (bay, horizontal and vertical position).

Not all of these characteristics are used for all modes. The position in a ship is only used for jumbos and deepseas, whereas a category number is only available for containers to be loaded into one of the jumbos or deepseas. For the third program these variables are extended with variables for the position in the stack (lane, row, position in row and height).

Input

Part of the input needed is fixed and put as constants to the program. It concerns the following variables:

- the arrival times of the jumbos and deepseas according to the fixed scheme and the maximum disturbance
- the number of jumbos and deepseas per period, and the number of other modalities (four: truck, shortsea/feeder, rail and barch)
- the distributions of the arrivals for the modes truck, shortsea/feeder, rail and barge during the week
- the period length (eg. three weeks)
- the sizes of the bays for jumbos and deepseas

Besides that, there are three input files. The first two files are '*jjJumbo.txt*' and '*ddDpsea.txt*'. Among other things, these files provide the following information:

- the number of bays for respectively a jumbo or a deepsea
- the number of quay cranes used to load and unload the ship under consideration
- the allocation of bays to the cranes. In other words, which bays are handled by which cranes?
- the bays. Every bay is named twice, once before unloading and once after loading. For the containers to be unloaded only the positions within the bay and their lengths are known. For the containers to be loaded, the category numbers are also known.
- the starting times of the cranes, relatively to the arrival time of the ship
- the number of hatches to be handled

The third input file to be used is the file '*flows.txt*'. This file provides a $(n \times n)$ -matrix, where n is the number of jumbos plus the number of deepseas plus four (which is the number of other modes). The matrix contains the flows calculated using Excel (see the chapter flows).

Procedures used

The program starts reading in the necessary input. The first procedure to be used is *ReadInput*. This procedure determines the order of the containers to be handled per jumbo/deepsea per crane. The procedure uses the input files '*jjJumbo.txt*' and '*ddDpsea.txt*'. A few steps are taken to obtain the desired result:

1. First of all the number of bays and the number of cranes per jumbo/deepsea are read.
2. The input files contain information about which bays are handled with which cranes. Together with the containers per bay this gives the containers per crane (The containers to be unloaded are unloaded from left to right and downward. The containers to be loaded into the ship are handled from left to right and upward). The containers are (ordered) written to output files '*Jkraani.txt*' for jumbos and '*Dkraani.txt*' for deepsea ships, where i stands for the crane number. For every container four values are written to file. First of all a -1 for containers to be

unloaded and a category number for containers to be loaded. Furthermore, the position in the ship is written to file: bay number, horizontal and vertical position within the bay.

3. Further, the starting times of the cranes with respect to the arrival times of the jumbos, respectively deepseas are read from the input files.
4. The number of hatches per bay is determined.

The next procedure is *ReadFlows*. This procedure reads in the matrix from the file '*flows.txt*'.

The next two procedures are almost the same (maybe we can make one procedure of them). The procedure *DetermineArrivals* determines the arrival times of the jumbos and deepseas that, according to the fixed scheme, should arrive in the first period. The procedure *NextArrivals* replaces these times, whenever the containers for a ship are generated, by the arrival times in the next period.

The procedure *Quicksort* is almost identical to the standard Quicksort procedure known in literature (ref ?). The only difference is that instead of wanting to know the arrival times in the right order, we want to know the ship numbers in the right order. So, we added an extra array which indicates where the ship that arrives as the x-th ship, was in the original order. By the way, this order is: first all jumbos in a period in the right order, then all deepsea ships.

The last two procedures are the most important ones: *GenerateContainers* and *WriteContainers*. The first procedure consists of two parts. The first part concerns the containers that are either brought or taken away (or both) with one of the jumbos or deepseas. The second part handles with the containers that are brought as well as taken away with one of the other modes.

In these procedures two time intervals are considered. The first interval which we will call the generation-interval is the interval used to generate the containers in the procedure *GenerateContainers*. The second interval is the so-called writing-interval, which is the period over which the containers are written to the output files in the procedure *WriteContainers*.

Starting with the first part of *GenerateContainers*, the procedure begins searching for the first ship to arrive in a specified time interval. If no such ships are found the procedure ends, but what happens if there is a ship that arrives during the time interval? First of all the ship is written to an output file. What statistics are outputted will be explained later. Then a new queue is created and there are drawn precisely as many moments in time as the total amount of containers to be loaded and the number of containers to be unloaded by this ship. These moments in time are uniformly distributed on the interval between one and twenty-three hours after the arrival of the ship. These times are added to the queue, where the position of entering the queue is based on the time (The times are sorted from low to high).

The second step is to add the data in the '*Jkraani.txt*' and the '*Dkraani.txt*' files. As stated before, these files are lists with containers in the same order as they are handled. The only thing to do is to pick one of the available cranes (keeping in mind their starting times). So for each moment a

crane is randomly chosen and the statistics of the next container for that crane are added to the next time in the queue.

At the same time a few other statistics are added. For this cause a distinction between containers to be unloaded and containers to be loaded has to be made. If a container is to be unloaded with the current ship, it means that this container doesn't already exist. So, a new container is defined. This means the container is given an (unique) identification number. Furthermore, the statistics from the previous paragraph refer to the moment of unloading. Thus, the ship which unloads the container is known, as are the position in the ship and the time the unloading takes place. Then, based on the flows, an object used to take away the container is chosen. If this object is one of the jumbos or deepseas the container is moved to the queue *Unload*, which contains all containers for which the unload statistics are known, but not the load statistics. If the object is an object from one of the other modes, then a moment for loading is chosen according to the right distribution. Now, all data are known for this container, so it is moved to the queue *Both*.

If the container is to be loaded by the current ship, the container might already exist in the queue *Unload*. This is the case when the container is also brought by one of the modes jumbo or deepsea. Only exception to this rule takes place in the beginning of the simulation, because there are no ships before time 0 and some of the containers loaded in week 1 are unloaded before this time by jumbo or deepsea. If this happens, the container is not defined. Therefore this week can't be used to check the number of containers. Besides, the container is also not known when it came with one of the other modes.

So, when a container is to be loaded by the current ship, first thing to do is to determine the object which brought the container. If this was one of the jumbos or deepseas, the queue *Unload* is searched for a container that has unload-object and load-object equal to the container under consideration. If found, the data about the moment of loading is added to this container and it is moved from *Unload* to *Both*. If not, the container is, as stated above, "thrown away". Whenever the container is brought by one of the other modes, the container doesn't already exist. This means we have to define a new container. All about the moment of loading is known, as is the mode which unloads the container. Last thing to determine in this case is the moment of loading. This happens based on the respective distribution. Then the container can be written to the queue *Both*.

The second part is only executed whenever the next writing-interval starts in a new week (because of the distributions being on weekly basis). Then the containers between the other modes are determined for this new week. This happens by drawing two numbers out of the arrival distributions belonging to this modes. The moment of unloading always takes place in this week. When the moment of loading comes later on in the week, both moments lie in the same week. If not, loading happens in the next week.

All there is to say yet, is that the containers are written to the queues, sorted on the time their next move takes place. The explanation above can be reflected by the next piece of pseudo-code:

Part I

for all jumbos and deepseas
if arrival during generation-interval

```

    determine moments for all moves coming with this object
  for every container
    determine a quay crane handling the container
    add data from 'Jkraani.txt' or 'Dkraani.txt'
    if unloading
      give ID
      data above concern moment of unloading
      determine loading-object
      if jumbo or deepsea
        add container to queue Unload
      else
        determine moment of loading
        add container to queue Both
    if loading
      determine unloading-object
      if jumbo or deepsea
        search for container in queue Unload
        if found
          add data loading moment
          move container from queue Unload to
queue Both
        if not found
          remove container
      else
        determine moment of unloading
        add container to queue Both

```

Part II

```

if next writing-interval is the beginning of a new week
  draw moment of unloading and moment of loading
  if moment of loading < moment of unloading
    moment of loading is one week later
  add container to queue Both

```

The last procedure to be discussed is the procedure *WriteContainers*. This procedure is used to write all container moves, occurring during the writing-interval to output files. As stated above the containers are in chronological order. A counter goes from the start of the interval to the end. Whenever a moves' time is equal to the counter, the move is written to file. There are two possibilities. The move is into or out of the stack. If the container goes into the stack this means the container is still in the queue *Both* and the moment of loading is yet to come. After writing the data to an output file (the data will be explained hereunder), the container is moved from *Both* to another queue called *Load*. So, in this queue are all containers that are currently in the stack, they are unloaded but not yet loaded. If the container comes out of the stack that means it can be found in this last queue. After writing the output to file, the container is then removed from the program.

This program can be summarised by the next pseudo-code:

```

define first generation-interval and first writing-interval.
ReadInput
ReadFlows
DetermineArrivals (first period)
Quicksort
GenerateContainers

```

```

NextArrivals
WriteContainers

repeat
  next generation-interval
  next writeing-interval
  Quicksort
  GenerateContainers
  NextArrivals
  WriteContainers
until end of simulation period

```

For the first generation-period we have chosen for the first nine days of the first three-week period. This generation-period is, on the one hand, long enough to ensure that all data of a container is known at the moment of unloading the container. On the other hand, this period is as short as possible to delimit the amount of time needed for the program to run. The first writing period includes all (loading) moves before time 0. After the first time these periods are always the day after the previous period. So, the second generation-period is day 10, whereas the second writing period is day 1, and so on.

Output

The last issue to discuss for this program is its output. As said before, the output consists of a file with jumbo and deepsea arrivals and several files with container moves. As for the ship arrivals, only variables written to the output file are their numbers and their arrival times. For the container moves a few things are written to file. First of all, is it a move into or out of the stack? This is denoted by -1 for unloading and +1 for loading. Then the containers ID-number and category are named. Furthermore, the positions in the objects are written to file, as are the objects that load or unload the container and the moments these actions take place.

So, part of the output could be:

```

-1 107643 11 6 13 10 6 6 6 7 5235338 1 5776780
-1 118210 40 -17360 -29248 23485 10194 -26616 21829 10 5235343 9 5797740
1 106385 28 3 11 6 14 13 3 9 4999411 7 5235352
1 104379 20 -30972 29445 9 0 20224 -27120 11 4803339 9 5235360
1 104596 40 -30972 -7163 30 0 23040 -27222 11 4832362 9 5235384
-1 105688 16 19 11 15 1 8 9 7 5235402 1 5783098
1 98347 40 8 9 3 -24571 3863 -27121 3 4917717 9 5235403
-1 0 0 -30972 18949 8 0 1792 -27634 7 5235407 0 771751936
1 106386 28 24 16 8 14 11 4 10 5063034 7 5235430
-1 107047 6 9 5 3 5 18 12 7 5235485 1 5725871

```

The selected line represents the stacking of container 105688. It comes with deepsea 4 (this is the 7) and leaves with jumbo 1 (1). Its category is 16 for jumbo ships. Furthermore, the container is in bay 19 position (11,15) in the deepsea and in bay 1 position (8,9) in the jumbo.

Some of the variables above aren't used for all modes. Because it is easier to write all containers to the output files in the same format, we have chosen to do so. Besides, the unnecessary variables don't cause any problems, they're just not used.

4.1.2. Test program

As mentioned above, this program is used to check the validity of the first program and the input of the first program. We will briefly describe all checked variables.

- First of all, the test program counts the number of containers handled. Actually it counts the number of containers entering the stack and the number of containers leaving the stack. These numbers should be approximately the same. Because of the disturbances, which might cause a container to enter or leave the stack a week earlier or a week later, these numbers are unlikely to be exactly the same. Furthermore the numbers should be approximately equal to the number of periods times 62000 (for the scenario “High”). Better yet, the program checks all possible flows between all modes. Per period they should be approximately equal to the numbers mentioned in the table.
- Secondly, the program checks whether the distribution of truck arrivals corresponds with the input distribution. In the same way the (random) distribution of the arrivals of the modes shortsea/feeder, rail and barge is checked.
- The time a container spends in the stack is the next issue to be considered. Of course, negative times shouldn't be possible, the maximum time shouldn't be too long and the average should be approximately equal to 3.9 days. The program determines the minimum, maximum and average time a container spends in the stack for all possible flows.
- The arrival times of the ships are tested. Are the disturbances indeed uniformly distributed on the interval between -0.5 days and 0.5 days?
- Finally, the program also calculates the maximum number of containers in the stack at a certain moment. In this manner, the minimum stack capacity needed can be determined.

For some of the things mentioned above, a difference would be very strange. For instance, the disturbances for the ship arrivals are drawn from the specified distribution. It would, therefore, be very awkward if the final distribution shouldn't match the input distribution.

4.1.3 Execution of stacking algorithms

The third program is the program we are going to repeat several times with different stacking algorithms. This program uses the input generated by the first program. Combined with a specified stacking algorithm and the classifications of the jumbo and deepsea categories, the simulation can be executed. The lengths for containers, that are brought and taken away by one of the other modes, are determined by choosing one of the possible lengths in approximately the same proportions as they appear to be for jumbo and deepsea ships. The program calculates certain statistics that could define a better or a worse algorithm. After running the program for all stacking algorithms we hope to be able to classify the tested algorithms.

Understanding the program is very simple. The container moves are read sequentially from the files. When the next move is a move into the stack the stacking algorithm is used to determine a position in the stack. If it concerns a move out of the stack the container is searched in the stack and leaves it.

There are a few basic principles for all stacking algorithms. First of all, containers of different sizes can't be placed on top of each other. Whenever there are reshuffles the containers to be reshuffled are positioned as close as possible to their current position.

The comparison of the stacking algorithms is based on the next statistics.

- Number of necessary reshuffles. This means only reshuffles that can't be avoided using the specific stacking algorithm. The number of reshuffles during repositioning is counted, but doesn't affect the choice of algorithm.
- Total distance. This means that an algorithm is better whenever the containers are closer to the point where they leave the stack. Besides, a position in the lane opposite to, for instance, the ship with which it is taken away, is better than the container being stacked in another lane. Therefore, it is important to know where the different modes arrive with respect to the stack.
- Peeks in the number of containers for any ASC. The number of containers per ASC in a specific time period is restricted by the capacity of this ASC. Therefore, the peeks should be measured. Whenever the number of containers during the peek is too high, the stacking algorithm isn't usable. The program measures the number of containers for every ASC for every quarter of an hour.

Last of all (but this requires running the programs a few times), the stacking algorithms can be compared based on their reaction to disturbances. This means that the simulation is run again by adding a probability of changes in the container specifications after the moment of stacking. This will result in a container being in the wrong spot. How does the algorithm cope with that? It is also possible to determine the reaction of an algorithm to allowing the possibility of repositioning during idle time.

5. Results of the test program

As said before, the test program is used to test the generation program. It assures that the basic variables are indeed what they are supposed to be. Hereunder, we will shortly discuss some of these variables, starting with the generated number of containers. Five subsequent periods of three weeks have been generated, whereas all outcomes are based on the last four periods. This is because of the first period (the null-period) not being complete.

The results concerning the flows between the objects/modalities and the results concerning the time a container spends in the stack are only presented for the case with simulation length equal to four three-week periods and the new distributions for shortsea/feeder, rail and barge.

5.1 The generated number of containers

In the Flows section a scenario is presented with 37,200 container visits per period. So, for this scenario, the number of containers during the four-period simulation length should be approximately four times this value, which is 148,800. The number of containers is calculated twice, once based on the number of containers that are loaded during the four periods and once based on the containers unloaded during the same period. The numbers are respectively 149,286 and 149,585. The next table presents the flows during the four periods. Each time, the first number is the number according to the pre-specified flows, the second number is the number based on the number of unloading operations and the third number is the number of containers handled according to the number of loading operations.

From/to	J1	J2	J3	D1	D2	D3	D4	Tr	SS/F	Ra	Li
J1	0	0	0	0	2332	0	0	588	5284	1408	2040
	0	0	0	0	2332	0	0	588	5284	1408	2040
	0	0	0	0	2332	0	0	588	5284	1408	2040
J2	0	0	0	0	0	2332	0	588	5284	1408	2040
	0	0	0	0	0	2332	0	588	5284	1408	2040
	0	0	0	0	0	2332	0	588	5284	1408	2040
J3	0	0	0	2332	0	0	2332	444	3960	1052	1532
	0	0	0	2332	0	0	2332	444	3960	1052	1532
	0	0	0	2482	0	0	2332	442	3916	1061	1514
D1	1796	0	0	0	1344	0	0	360	1948	524	756
	2015	0	0	0	1476	0	0	401	2139	585	822
	1796	0	0	0	1344	0	0	360	1951	524	756
D2	0	3588	0	0	0	1344	0	180	976	264	376
	0	3588	0	0	0	1344	0	180	976	264	376
	0	3588	0	0	0	1344	0	180	976	264	376
D3	0	0	3588	0	0	0	1344	180	976	264	376
	0	0	3588	0	0	0	1344	180	976	264	376
	0	0	3588	0	0	0	1344	180	976	264	376
D4	1792	0	0	1344	0	0	0	360	1952	524	756
	1792	0	0	1344	0	0	0	360	1952	524	756
	1792	0	0	1499	0	0	0	354	1904	523	741
Tr	584	584	584	368	368	368	368	0	3868	0	0
	600	584	584	371	368	368	368	0	3864	0	0
	584	584	584	408	368	368	368	0	3864	0	0
SS/F	5160	5160	5160	2000	2000	2000	2000	3868	0	7504	10940
	5213	5160	5160	2023	1968	2000	2000	3864	0	7500	10932
	5160	5160	5160	2132	2000	2000	2000	3869	0	7475	10959
Ra	1396	1396	1396	540	540	540	540	0	7508	0	0
	1416	1396	1396	541	540	540	540	0	7500	0	0
	1396	1396	1396	599	540	540	540	0	7499	0	0

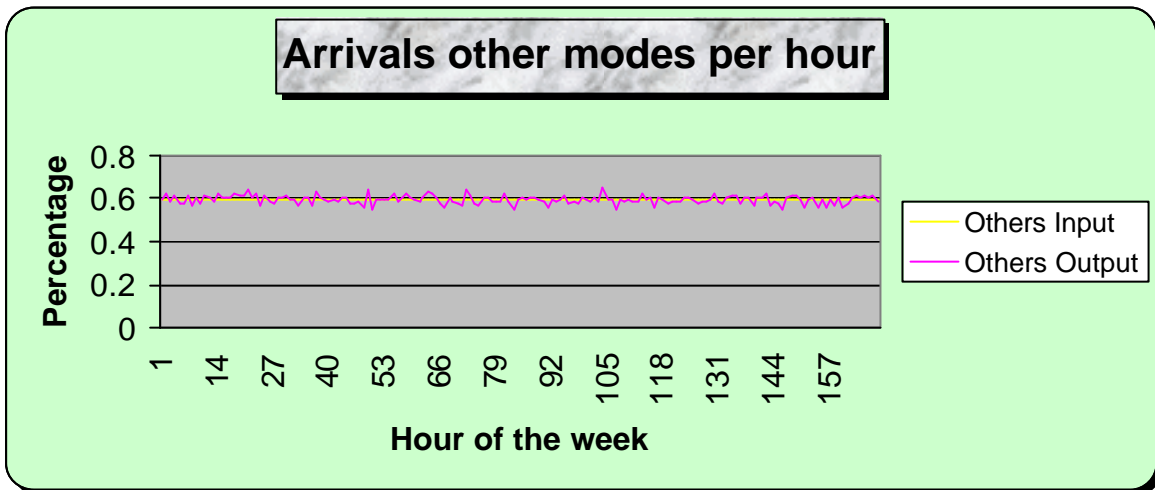
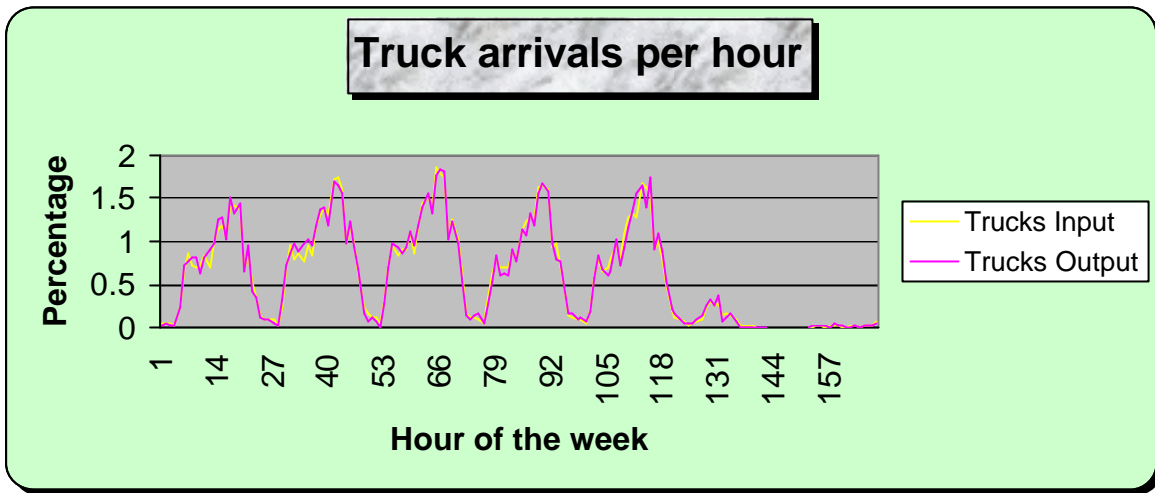
Li	2032	2032	2032	788	788	788	788	0	10944	0	0
	2063	2032	2032	786	778	788	788	0	10944	0	0
	2032	2032	2032	873	788	788	788	0	10953	0	0

This table suggests that the number of containers generated is close enough to the theoretical number. As is stated in the description of the test program the numbers are not expected to be exactly equal. Differences are caused by the existence of the disturbances in ship arrivals and the 'randomness' of the shortsea/feeder, truck, rail and barge arrivals. Note that all deepsea and jumbo ships always leave during the same period in which they arrive except for the first deepsea. This is incurred by the fixed arrival pattern of these ships and the maximum disturbances in arrivals being -0.5 or 0.5 days. The first deepsea will arrive precisely at the beginning of a new period according to the fixed scheme. Including the disturbances, it is then possible for this ship to arrive somewhere during the last twelve hours of the previous period, which will cause a deviation from the expected number of container handlings during that period. Note that this number will increase whenever the first deepsea in the period after the simulation stops is early (it is partly handled within the simulation period), whereas it will decrease whenever the first deepsea in the first period of the simulation is early (it is partly handled before the beginning of the simulation period).

The numbers of arrivals for shortsea/feeder, truck, rail and barge are in principle the same for every new period. It is, however, possible that these numbers change. This is caused by the way in which those container moves are generated. The arrival moments are picked randomly within a week. It is then possible that the time a container has to leave the stack comes before the moment at which it arrives. However, a container has to enter the stack at least three hours before it leaves the stack. Whenever this is not the case in first instance (within the program), the container is to leave the stack exactly one week later (except for the containers that are brought by shortsea/feeder, truck, rail or barge and taken away by jumbo or deepsea. In that case the arrival moment is exactly one week earlier).

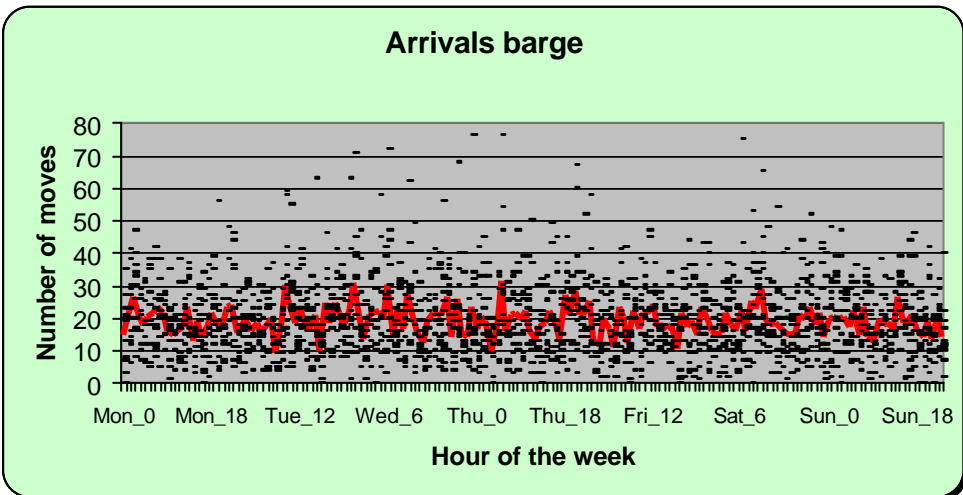
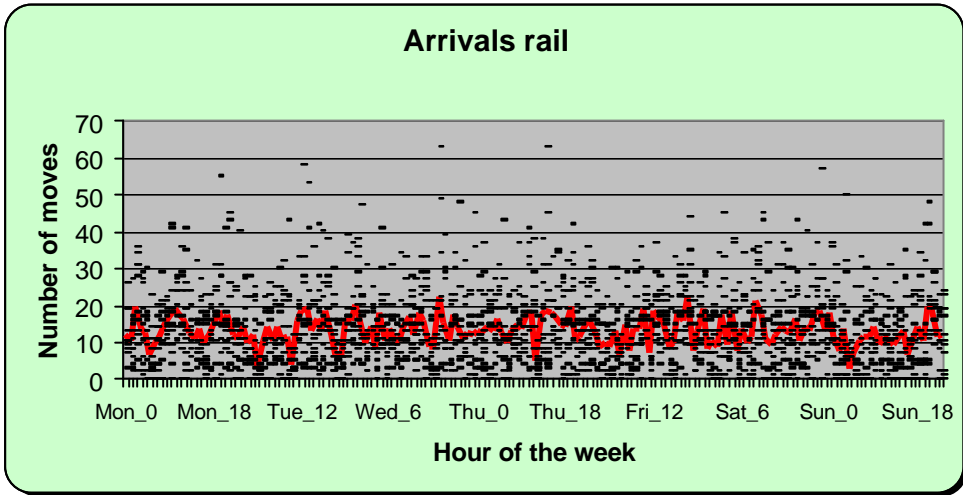
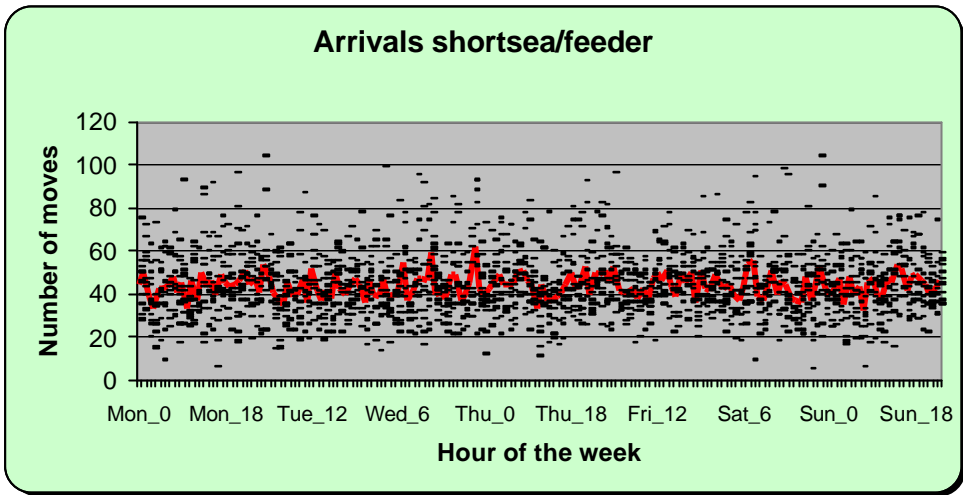
5.2 Arrivals of trucks and the other modalities

The test program determined the distribution of the arrivals over the hours of the week. Using Microsoft Excel the following graphs are constructed. In the first graph the percentages of truck arrivals over the hours in a week are given (based on the four three-week periods). There are two lines, one displaying the predetermined percentages and one displaying the generated percentages. The second graph is the same, but for the other modalities, for which the predetermined percentages are assumed to be constant over the week (horizontal line).



The graphs show that the generated containers do (approximately) follow the specified distributions.

The post-simulation graphs of the new arrival distributions for the modalities shortsea/feeder, rail and barge can also be shown. Those distributions will never be the same as the distributions in section 3.1.3. However, they have to show a similar pattern.



5.3 The time a container spends in the stack

The average time a container spends in the stack is approximately 3.73 days. The maximum time a container spends in the stack is about 7.76 days, whereas the minimum time is three hours and seven seconds.

The maximum number of positions taken in the stack at any moment in time is calculated to be 13,382. This is based on the supposition of a 20 ft. container taking one position, a 40 ft. container two and a 45 ft. container three. This means that a stack with lower capacity than 13,382 will never work. In fact, to assure for open places in the stack when needed, the stack's capacity should be higher than 13,382.

Conclusion

The findings above indicate that the program meets the (modified) specifications.

6. References

Celen H.P., Slegtenhorst R.J.W., Van der Ham R. TH., Nagel A., Van den Berg J., De Vos Burchart R., Evers J.J.M., Lindeijer D.G., Dekker R., Meersmans P., De Koster M.B.M., Van der Meer J.R., Carlebur A.F.C., Nooijen F.J.A.M., (1997) **FAMAS-NewCon Definitiestudie**, CTT publicatiereeks 32, ISBN 90-76091-36-6.

Meersmans P.J.M., (1999) **Deadlock free scheduling of stacking cranes and automated guided vehicles**, Working paper Econometric Institute EUR.

Van der Meer J.R., De Koster R., Vis I.F.A., Meersmans P.J.M., Dekker R., (1999) **Containerterminals in het Famas NewCon concept**, te verschijnen Eburon Delft.

Meersmans, P.J.M., Vis, I. De Koster, R and Dekker, R. (1999), **FAMAS-NewCon: beschrijving van het korte-termijn stack model**, (in wording).

Cooper, D. (1987), **Condensed Pascal**, W.W. Norton & Company

Upward Systems (1994), **Must simulation software: User and reference manual**

Appendix I - Categories

Containers, that are to be loaded by one of the modes jumbo or deepsea can be divided into several categories. There are 45 possible categories for jumbo ships and 90 for deepseas.

The division of containers to be loaded into a jumbo into 45 categories is as follows:

20 ft. containers			
Weight \ Destination	A	B	C
Light	25	3	5
Medium	7	9	11
Heavy	13	15	17
Very heavy	19	21	23

40 ft. containers			
Weight \ Destination	A	B	C
Light	2	4	6
Medium	8	10	12
Heavy	14	16	18
Very heavy	20	22	24

45 ft. containers			
Weight \ Destination	A	B	C
Light	40	41	42
Medium	43	44	45
Heavy	46	47	48
Very heavy	49	50	51

Reefers			
Length \ Destination	A	B	C
20 ft.	30	31	32
40 ft.	33	34	35
45 ft.	36	37	38

In one table:

Destination Length (ft.)	A			B			C		
	20	40	45	20	40	45	20	40	45
Light	25	2	40	3	4	41	5	6	42
Medium	7	8	43	9	10	44	11	12	45
Heavy	13	14	46	15	16	47	17	18	48
Very heavy	19	20	49	21	22	50	23	24	51
Reefers	30	33	36	31	34	37	32	35	38

The categories for a deepsea ship are:

20 ft. containers						
Weight \ Destination	A	B	C	D	E	F
Light	49	3	5	7	9	11
Medium	13	15	17	19	21	23
Heavy	25	27	29	31	33	35
Very heavy	37	39	41	43	45	47

40 ft. containers						
Weight \ Destination	A	B	C	D	E	F
Light	2	4	6	8	10	12
Medium	14	16	18	20	22	24
Heavy	26	28	30	32	34	36
Very heavy	38	40	42	44	46	48

45 ft. containers						
Weight \ Destination	A	B	C	D	E	F
Light	50	51	52	53	54	55
Medium	56	57	58	59	60	61
Heavy	62	63	64	65	66	67
Very heavy	68	69	70	71	72	73

Reefers						
Length \ Destination	A	B	C	D	E	F
20 ft.	80	83	86	89	92	95
40 ft.	81	84	87	90	93	96
45 ft.	82	85	88	91	94	97

Together:

Destination	A			B			C		
	20	40	45	20	40	45	20	40	45
Light	49	2	50	3	4	51	5	6	52
Medium	13	14	56	15	16	57	17	18	58
Heavy	25	26	62	27	28	63	29	30	64
Very heavy	37	38	68	39	40	69	41	42	70
Reefers	80	81	82	83	84	85	86	87	88

Destination	D			E			F		
	20	40	45	20	40	45	20	40	45
Light	7	8	53	9	10	54	11	12	55
Medium	19	20	59	21	22	60	23	24	61
Heavy	31	32	65	33	34	66	35	36	67
Very heavy	43	44	71	45	46	72	47	48	73
Reefers	89	90	91	92	93	94	95	96	97

Appendix II - Input variables

This appendix contains the values for the input variables for the speed and acceleration of ASCs, the sizes of stack positions, the depth of the reefer platform and the time ASCs need for positioning when picking up or putting down a container.

Sizes of stack positions

- Length 6.4000 m
- Width 2.9900 m
- Height 2.5904 m

ASC movement

- Length speed: 4.0 m/s acceleration: 0.3 m/s²
- Width speed: 0.8 m/s acceleration: 0.3 m/s²
- Height (full) speed: 0.6 m/s acceleration: 0.3 m/s²
- Height (empty) speed: 1.0 m/s acceleration: 0.3 m/s²

Positioning times

- Picking up of a container 3.0 s
- Putting down of a container 3.0 s

Depth of reefer platform 10 ft.

Appendix III – Flows (input)

This appendix contains the flows per three-week period as they are used (scenario ‘Low’) or can be used (scenario ‘High’) in the generation program. These flows are based on the modal split and on the restriction of an average dwell time of 3.9 days.

Scenario ‘Low’

We have used the following abbreviations:

- F / T From / To
The objects (or modalities) that deliver the container to the terminal are in rows.
The objects (or modalities) that take the container away from the terminal are in columns.
- J Jumbo ship (1,2,3)
- D Deepsea ship (1,2,3,4)
- Tr. Truck
- SS/F. Shortsea and feeder ships
- Ra. Rail
- Ba. Barge

F / T	J 1	J 2	J 3	D 1	D 2	D 3	D 4	Tr.	SS/F.	Ra.	Ba.
J 1	0	0	0	0	583	0	0	147	1321	352	510
J 2	0	0	0	0	0	583	0	147	1321	352	510
J 3	0	0	0	583	0	0	583	111	990	263	383
D 1	449	0	0	0	336	0	0	90	487	131	189
D 2	0	897	0	0	0	336	0	45	244	66	94
D 3	0	0	897	0	0	0	336	45	244	66	94
D 4	448	0	0	336	0	0	0	90	488	131	189
Tr.	146	146	146	92	92	92	92	0	967	0	0
SS/F.	1290	1290	1290	500	500	500	500	967	0	1876	2735
Ra.	349	349	349	135	135	135	135	0	1877	0	0
Ba.	508	508	508	197	197	197	197	0	2736	0	0

Scenario ‘High’

The flows for the scenario with around 3,1 million container visits per year have been split up into three tables. This is due to the fact that in this case there are fifteen jumbo arrivals and twenty deepsea arrivals per period. The abbreviations are the same as above.

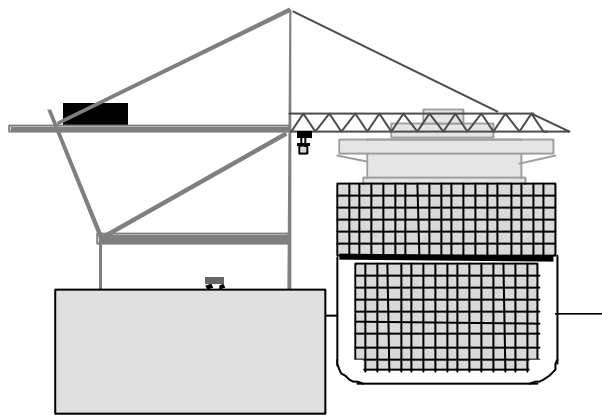
	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13
J1	0	0	0	0	0	0	0	0	0	0	0	0	0
J2	0	0	0	0	0	0	0	0	0	0	0	0	0
J3	0	0	0	0	0	0	0	0	0	0	0	0	0
J4	0	0	0	0	0	0	0	0	0	0	0	0	0
J5	0	0	0	0	0	0	0	0	0	0	0	0	0
J6	0	0	0	0	0	0	0	0	0	0	0	0	0
J7	0	0	0	0	0	0	0	0	0	0	0	0	0
J8	0	0	0	0	0	0	0	0	0	0	0	0	0
J9	0	0	0	0	0	0	0	0	0	0	0	0	0
J10	0	0	0	0	0	0	0	0	0	0	0	0	0
J11	0	0	0	0	0	0	0	0	0	0	0	0	0
J12	0	0	0	0	0	0	0	0	0	0	0	0	0
J13	0	0	0	0	0	0	0	0	0	0	0	0	0
J14	0	0	0	0	0	0	0	0	0	0	0	0	0
J15	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	269	179	135	45	0	0	0	0	0	0	0
D2	0	0	0	269	179	135	0	0	0	0	0	0	0
D3	0	0	0	269	269	179	45	0	0	0	0	0	0
D4	0	0	0	0	269	269	135	45	0	0	0	0	0
D5	0	0	0	0	0	269	179	135	45	0	0	0	0
D6	0	0	0	0	0	0	269	179	135	0	0	0	0
D7	0	0	0	0	0	0	269	269	179	45	0	0	0
D8	0	0	0	0	0	0	0	269	269	135	45	0	0
D9	0	0	0	0	0	0	0	0	269	179	135	45	0
D10	0	0	0	0	0	0	0	0	0	269	179	135	0
D11	0	0	0	0	0	0	0	0	0	269	269	179	45
D12	0	0	0	0	0	0	0	0	0	0	269	269	135
D13	0	0	0	0	0	0	0	0	0	0	0	269	179
D14	0	0	0	0	0	0	0	0	0	0	0	0	269
D15	45	0	0	0	0	0	0	0	0	0	0	0	269
D16	135	45	0	0	0	0	0	0	0	0	0	0	0
D17	179	135	45	0	0	0	0	0	0	0	0	0	0
D18	269	179	135	0	0	0	0	0	0	0	0	0	0
D19	269	269	179	45	0	0	0	0	0	0	0	0	0
D20	0	269	269	135	45	0	0	0	0	0	0	0	0
Tr.	146	146	146	146	146	146	146	146	146	146	146	146	146
SS/F.	1290	1290	1290	1290	1290	1290	1290	1290	1290	1290	1290	1290	1290
Ra.	349	349	349	349	349	349	349	349	349	349	349	349	349
Ba.	508	508	508	508	508	508	508	508	508	508	508	508	508

	J 14	J 15	D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9	D 10	D 11
J 1	0	0	0	0	225	225	225	83	50	50	0	0	0
J 2	0	0	0	0	0	0	225	225	83	83	50	0	0
J 3	0	0	0	0	0	0	0	225	225	225	83	50	0
J 4	0	0	0	0	0	0	0	0	225	225	225	83	50
J 5	0	0	0	0	0	0	0	0	0	0	225	225	83
J 6	0	0	0	0	0	0	0	0	0	0	0	225	225
J 7	0	0	0	0	0	0	0	0	0	0	0	0	225
J 8	0	0	0	0	0	0	0	0	0	0	0	0	0
J 9	0	0	0	0	0	0	0	0	0	0	0	0	0
J 10	0	0	0	0	0	0	0	0	0	0	0	0	0
J 11	0	0	50	0	0	0	0	0	0	0	0	0	0
J 12	0	0	83	50	0	0	0	0	0	0	0	0	0
J 13	0	0	225	83	50	50	0	0	0	0	0	0	0
J 14	0	0	225	225	83	83	50	0	0	0	0	0	0
J 15	0	0	0	225	225	225	83	50	0	0	0	0	0
D 1	0	0	0	0	100	100	34	34	34	34	0	0	0
D 2	0	0	0	0	0	100	100	34	34	34	34	0	0
D 3	0	0	0	0	0	0	100	100	34	34	34	34	0
D 4	0	0	0	0	0	0	0	100	100	34	34	34	34
D 5	0	0	0	0	0	0	0	0	100	100	34	34	34
D 6	0	0	0	0	0	0	0	0	0	100	100	34	34
D 7	0	0	0	0	0	0	0	0	0	0	100	100	34
D 8	0	0	0	0	0	0	0	0	0	0	0	100	100
D 9	0	0	0	0	0	0	0	0	0	0	0	0	100
D 10	0	0	0	0	0	0	0	0	0	0	0	0	0
D 11	0	0	0	0	0	0	0	0	0	0	0	0	0
D 12	45	0	0	0	0	0	0	0	0	0	0	0	0
D 13	135	45	0	0	0	0	0	0	0	0	0	0	0
D 14	179	135	34	0	0	0	0	0	0	0	0	0	0
D 15	369	179	34	34	0	0	0	0	0	0	0	0	0
D 16	269	269	34	34	34	0	0	0	0	0	0	0	0
D 17	0	269	34	34	34	34	0	0	0	0	0	0	0
D 18	0	0	100	34	34	34	34	0	0	0	0	0	0
D 19	0	0	100	100	34	34	34	34	0	0	0	0	0
D 20	0	0	0	100	100	34	34	34	34	0	0	0	0
Tr.	146	146	92	92	92	92	92	92	92	92	92	92	92
SS/F.	1290	1290	500	500	500	500	500	500	500	500	500	500	500
Ra.	349	349	135	135	135	135	135	135	135	135	135	135	135
Ba.	508	508	197	197	197	197	197	197	197	197	197	197	197

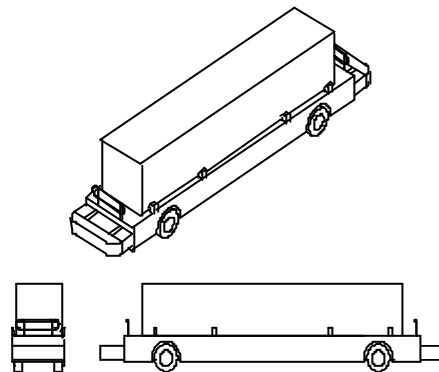
	D 12	D 13	D 14	D 15	D16	D 17	D 18	D 19	D 20	Tr.	SS/F.	Ra.	Ba.
J 1	0	0	0	0	0	0	0	0	0	130	1165	310	450
J 2	0	0	0	0	0	0	0	0	0	143	1269	340	495
J 3	0	0	0	0	0	0	0	0	0	134	1196	315	460
J 4	50	0	0	0	0	0	0	0	0	130	1165	310	450
J 5	83	50	0	0	0	0	0	0	0	143	1269	340	495
J 6	225	83	50	0	0	0	0	0	0	134	1196	315	460
J 7	225	225	83	50	50	0	0	0	0	130	1165	310	450
J 8	0	225	225	83	83	50	0	0	0	143	1269	340	495
J 9	0	0	225	225	225	83	50	0	0	134	1196	315	460
J 10	0	0	0	225	225	225	83	50	50	130	1165	310	450
J 11	0	0	0	0	0	225	225	83	83	143	1269	340	495
J 12	0	0	0	0	0	0	225	225	225	134	1196	315	460
J 13	0	0	0	0	0	0	0	225	225	130	1165	310	450
J 14	0	0	0	0	0	0	0	0	0	143	1269	340	495
J 15	0	0	0	0	0	0	0	0	0	134	1196	315	460
D 1	0	0	0	0	0	0	0	0	0	72	390	105	151
D 2	0	0	0	0	0	0	0	0	0	77	415	110	161
D 3	0	0	0	0	0	0	0	0	0	58	320	83	123
D 4	0	0	0	0	0	0	0	0	0	63	341	91	133
D 5	34	0	0	0	0	0	0	0	0	72	390	105	151
D 6	34	34	0	0	0	0	0	0	0	77	415	110	161
D 7	34	34	34	0	0	0	0	0	0	58	320	83	123
D 8	34	34	34	34	0	0	0	0	0	63	341	91	133
D 9	100	34	34	34	34	0	0	0	0	72	390	105	151
D 10	100	100	34	34	34	34	0	0	0	77	415	110	161
D 11	0	100	100	34	34	34	34	0	0	58	320	83	123
D 12	0	0	100	100	34	34	34	34	0	63	341	91	133
D 13	0	0	0	100	100	34	34	34	34	72	390	105	151
D 14	0	0	0	0	100	100	34	34	34	77	415	110	161
D 15	0	0	0	0	0	100	100	34	34	58	320	83	123
D 16	0	0	0	0	0	0	100	100	34	63	341	91	133
D 17	0	0	0	0	0	0	0	100	100	72	390	105	151
D 18	0	0	0	0	0	0	0	0	100	77	415	110	161
D 19	0	0	0	0	0	0	0	0	0	58	320	83	123
D 20	0	0	0	0	0	0	0	0	0	63	341	91	133
Tr.	92	92	92	92	92	92	92	92	92	0	4835	0	0
SS/F.	500	500	500	500	500	500	500	500	500	4835	0	9380	13675
Ra.	135	135	135	135	135	135	135	135	135	0	9385	0	0
Ba.	197	197	197	197	197	197	197	197	197	0	13680	0	0

Appendix IV – List of abbreviations and definitions

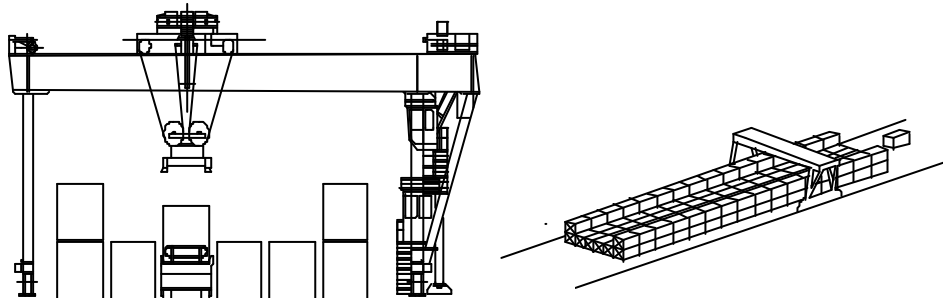
- TEU Twenty feet Equivalent Unit, meaning one 20 ft. container. A 40 ft. container equals two TEU.
- Reefer These are cooling containers. They require a connection to the electricity net, as well on board as in the stack.
- Stack The stack is the space where containers are (temporarily) stored. Most of the times there are special parts of the stack for reefer containers.
- QC Quay Crane: the crane that picks up containers from or loads containers to seaships



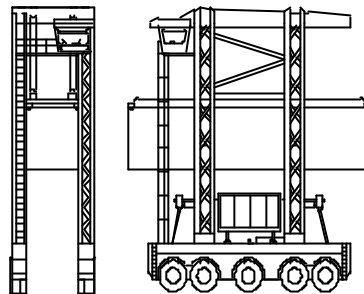
- AGV Automated Guided Vehicle: transports the containers from the stack to the quay and vice versa



- ASC Automatic Stacking Crane: is used to put containers into the stack and remove containers out of the stack



Spreader
SC The 'instrument' that is used by QCs and ASCs to pick up containers
Straddle Carrier. This vehicle is used to load containers onto trucks. The container fits within the legs of the SC. It can move over a pile of two or three containers.



MTS Multi Trailer System (manned). This is a vehicle with five trailers behind it. Is used at the DMU and for transport between stack and rail terminal.



Jumbo Very large container ship (> 8000 TEU).

Deapsea Container ship for intercontinental traffic (< 8000 TEU)

Shortsea Container ship for international traffic (for example Europe, Northern-Africa)

Feeder Container ship that supplies cargo for the larger ships (for example cargo from Hamburg for a ship to the Middle East)

Barge Ship for inland shipping of containers.

ITT Inter Terminal Transport. The transport of containers between the different service centres (for instance from the stack to the rail terminal).

Quay transport The transport between the stack and the quay cranes.

Category Group of containers that are mutually exchangeable, because of them having (approximately) the same weight, sizes and destination.

Reshuffle A container that is positioned (in the stack) on top of another container, which has to leave the stack earlier, so it has to be replaced.

Pre-positioning	A container in the stack is already shifted towards the transfer point, to make it possible to take it faster out of the stack at a later time
DMU	Delta Multi User terminal. This is the first terminal on the Maasvlakte. It is still manned.
DSL	Delta/Sea-Land terminal. This terminal is dedicated to the shipowner Sea-Land Inc. It is the first automated terminal in the world (AGVs and ASCs).
DDE	Delta Dedicated East. This is the most recent terminal on the Maasvlakte. The DDE is also dedicated to a consortium of shipowners.
Home	The Home terminal. These are the terminals in the older part of the Rotterdam harbours.