Belgian Journal of Operations Research, Statistics and Computer Science Vol. 40 (3-4) 00

Increasing lock throughput in an inland port through simulation-optimisation

Gerrit K. Janssens

Operations Management and Logistics
University of Limburg (LUC)
Universitaire Campus – Building D
B-3590 Diepenbeek, Belgium
e-mail: gerrit.janssens@luc.ac.be

Abstract

A dual lock is considered, in which vessels enter the port from a river and leave the port for the river. Vessels enter the lock when it is open and wait until the locking operation is finished to move forward. A limited number of vessels can be placed in the lock. From a static view, the assignment of locations within the lock is an optimisation problem of the type 'packing problem'. The decision however is not static but dynamic, as a trade-off must be made between waiting for a newly arriving vessel to enter the lock or starting the locking operation. In the former case the newly arriving vessels faces a longer waiting time; in the latter case all vessels already present in the lock face the longer waiting time. The arrivals of vessels at the lock can be described as a stochastic process, both in time as in size of the vessels. Input to the dynamic packing problem is generated by means of a real-life existing simulation model for vessel traffic on the river Scheldt in Belgium.

Keywords: simulation-optimisation, packing problem, maritime application.

1 INTRODUCTION

Inland shipping is an important industry in Western Europe. The location of a country like Belgium close to the mouths of the Scheldt, Rijn and Maas rivers, means that its system of waterways forms a vital link between the industrial areas in inland Western Europe and the ports of Antwerp and Rotterdam.

The port of Antwerp is one of the major ports in the world. It is an upstream urban port on the river Scheldt, nearly one hundred kilometers inland of Belgium. The access to the port goes through a number of locks.

Before entering Belgium, the river crosses the Netherlands. Together with the ports of Rotterdam, Bremen and Hamburg, the port of Antwerp has superior overland and inland waterway connections with the major industrial areas in Western Europe. Antwerp has excellent access by rail, barge and highway to points within the once-famous coal- and steel-producing industrial triangle from Dunkerque to the French Lorraine and the German Ruhr (Fleming 1997).

Inland shipping depends on the use of locks. They serve shipping operations by maintaining adequate depths on waterways. In some countries locks also serve water management purposes. Additionally, by maintaining water in canals at different desired levels, locks conserve large amounts of water that can be used for various purposes (Bolten 1980).

Any construction or major maintenance activities influence the vessel traffic on the river and, by this, the performance of the whole port system. A port simulation model can be used for determining the effects of changes in throughput and various operational, technological and investment options. Simulation methods were constructed to analyse the impact of changes in throughputs, in fleet composition, in vessel arrival patterns and in terminal layouts (Hayuth et al. 1994).

It has been recognised that simulation can serve as a decision support tool for port management. The model can calculate the port performance indicators, the critical indices and the facility utilisation factors (Hassan 1993). Lots of models deal with the operations of terminals, e.g. in Wadhwa (1992), Kuehn et al. (1994), Kondratowicz (1994). Recently a simulation model has been developed taking into consideration the interdependence between locks, bidirectional traffic, dual chamber facilities and budget limitations (Martinelli et al. 1993). A simulation model for the Illinois waterway locks system is described in Bandy (1996). A review on published papers on port terminal simulation is given in Blümel (1997) and Toluyev et al. (1996). More concrete applications are reported in Blümel and Novitsky (2000).

A traffic simulation model for the river Scheldt has been developed and is described in Thiers and Janssens (1998). The simulation model is the object of a study which has been ordered by the Belgian Waterways regarding the plans to build a second container quay on the river Scheldt just downstream (i.e. North of) the locks of Berendrecht-Zandvliet. For the planned quay, these manoeuvres have to take place on an area with very dense traffic. This would result in hindrance for the other traffic on the river. Opinions amongst officials were divided about this hindrance, resulting from future increasing traffic.

2 THE RIVER SCHELDT SIMULATION MODEL

The simulation study is based upon an earlier version of a simulation model, called the Western Scheldt Simulation model (Thiers and Janssens 1998). The model is both adapted and extended according to the objectives of this study.

The model is a *traffic* simulation model. This means that navigation is not approached from its technical side nut in terms of time taken by specific activities. Two types of activities are distinguished: planned activities and condition dependent activities. *Planned activities* appear in a logical sequence. Examples are: the generation of a vessel according to an inter-arrival time distribution, its arrival at destination in function of its speed, and tides. *Condition dependent activities* relate to queues. If a certain condition is satisfied, the activity is executed, else the vessel has to wait. Some examples include: control on the tide and weather conditions for the vessel, control on the availability of a pilot, and control on the availability of a lock.

The model simulates the traffic on the Western Scheldt, the channel between Ghent and Terneuzen, at the locks and the ports. Sailing of vessels and interaction between vessels and between a vessel and its environment are described in a mathematical model.

The model is deterministic. No stochastic elements are included during sailing and in the interaction between vessels. Stochastic input is provided, only for the generation of traffic.

The original mathematical traffic simulation model includes the maritime access to the port of Antwerp and the development of various simulations for sea traffic at the lock complex of Berendrecht-Zandvliet, the operational container quay and the planned container quay.

The main policy was to develop a simulation model of the maritime traffic on the access paths to the port of Antwerp but in such a way that it can serve as a reusable tool (expandable and maintainable) for port planning. Afterwards the model has served for reusable purposes, extending it for several other requests.

A first one relates to plans to construct a tunnel below the river Scheldt, downstream of Terneuzen. It was decided that the tunnel should be constructed by sinking prefabricated tunnel elements into the river bottom. In total, during nearly ten months the waterway would be considerably narrowed, forcing ships to slow down and keeping vessels of larger size from crossing or overtaking within that area.

A second extension relates to a pilot operations study. The model has been extended with the traffic to and from other Flemish seaports like Bruges, Ghent and Ostend, as the sea-pilot dispatching system involves all Flemish ports (Janssens, Thiers and Tielemans 1999).

A third one relates to a lock renovation project. All traffic passing through this lock needs to be diverted to other locks, causing queues and delays at the locks. Both travel times and distances increase for most vessels. For a study on the unavailability of the lock, the simulation model needs extension towards the structure of the right-bank harbour including channels, docks, bridges, harbour pilots and harbour tugboats.

The simulation model has been implemented in the Arena software. Visual components have been proved to be useful in building maritime traffic simulation models (Taeymans and Bouwman 1998). A visual component library has been developed at five layers. The approach of using a five-layer model development hierarchy leads to a nearly unlimited parameterisation of the simulation model.

Layers one and two define the infrastructure. Layer one deals with the general view of the infrastructure. Here a visual component library is used to construct the infrastructure network of the simulation model (e.g. waterways, locks, bridges, ..). In a second layer the characteristics of each component are added (e.g. width and depth of a segment of the waterway).

Layers three and four provide the functionalities related to the use of the infrastructure. In the third layer various routes are defined (e.g. potential waterways connecting origin A with destination B). In the fourth layer the traffic rules (for various transport modi) can be determined. In layer five the empirical data are filled in. They can be obtained from measurements, from databases or from other sources. The data include e.g. vessel arrival patterns, origin/destination matrices, service hours of bridges and locks etc.

3 LOCK OPERATIONS IN THE PORT MODEL

Vessels passing through a lock take more time than they would if the lock were not there. This delay consists both of the time waiting to enter the locks as well as the time spent within the lock during a cycle. Water management increases delays at locks. While vessel operators would like frequent lock cycles, water management requires the opposite.

Lock operations are described shortly. Vessels approaching the lock moor while waiting to enter the lock. If the lock is open, they enter and tie up inside the lock until all waiting vessels are entered or the lock is full. Afterwards the doors are closed and the water level within the lock is raised or lowered up to the level with the waterway on the opposite side. When the levels are equal, the doors open again and the vessels leave the lock. The lock can be cycled back in the other direction.

A lock complex can consist of one or more locks arranged in parallel. A lock operator will direct the vessels to a specific lock. The choice depends on: vessel and lock dimensions, lock availability, weather conditions, safety considerations and queue lengths. Normally vessels enter locks in their order of arrival. Exceptions can exist in order to pack the lock for maximum capacity.

Lock operations require managerial tactics. Bolten (1980) mentions some potential tactics as (1) using the smallest suitable lock for an arriving vessel; or (2) requiring a minimum queue length before recycling; or (3) setting a maximum waiting time for vessels during low traffic periods; or (4) limiting cycles during high tide periods.

Our lock operation model is embedded into the greater port traffic simulation model. Lock and vessel traffic characteristics and technical characteristics determine the model parameters. Two important characteristics of a lock complex are the lock capacity and the vessel delay time. The lock operation model simulates the passage of vessels over a predetermined period using normal lock operating rules and managerial tactics.

Lock capacity can be thought of in two ways: either it relates to the number of vessels which can pass the lock in a fixed time period (e.g. a week), or it relates to the static chamber capacity, i.e. the maximum number of vessels in the chamber. We concentrate on the latter case. With a known capacity the lock behaves as a queue with a complex behavior. According to Kooman and De Bruijn (1975), vessel delay time consists of three components: locking (time spent entering, exiting and waiting inside the lock chamber), waiting (time between arrival of the vessel at the lock complex and the time the lock becomes available for ships to enter) and delay (additional waiting time cannot enter the lock at first opportunity because the queue is too long).

4 CHARACTERISTICS OF VESSEL TRAFFIC

In order to describe the vessel traffic using the locks, a difference is made between vessels of various size classes and types. The arrival pattern changes during the week. Both characteristics of vessels and their arrival patterns are subject to seasonal variations. This should allow us to compute performance measures as: the mean delay cost per vessel and per hour, and the mean number of vessels required to fill a lock chamber (Kooman and de Bruijn, 1975).

Vessel arrivals with destination Antwerp are generated at both entry points of the model (A1-buoy and Steenbank), according to a non-stationary Poisson process. Arrival rates vary from hour to hour. Also they vary between days in the week and months in the year. This non-stationary behavior is modeled using a system of month-, day- and hour-indices. The time-dependent arrival rate for a specific vessel class is given by:

$$AR_{(hour)} = \frac{TNV}{365 * 24} * I_M * I_D * I_H ,$$

where $AR_{(hour)}$ = arrival rate (per hour)
TNV = total number of vessels (per year)
I_M = monthly index
I_D = daily index
I_H = hourly index.

This arrival rate is the parameter of a Poisson distribution from which inter-arrival times are generated, which follow an exponential distribution. The inter-arrival times are calculated per location per type of vessel. Once a vessel generated, its class is being determined per location. The shares of size classes within a vessel type is calculated based on data from 1995 as these data are also used for validation of the model. For simulation of future situations this information is obtained from forecasts.

At generation of a trip all attributes are assigned and checks are made whether the vessel can start its journey. These controls include conditions on tide and weather. If conditions are not satisfied, the vessel is put in a queue and conditions are re-checked every five minutes.

The port authorities of Antwerp and Rotterdam draw up forecasts on a periodical basis. The forecasts for the sea-borne traffic in the port of Antwerp are carried out by the Study Centre for the Expansion of Antwerp (SEA) of the Antwerp Port Authority. Forecasts were made per category and size class of vessel, for multiple years in the future. They take into account the effects of expected infrastructure changes (Coeck et al. 1996).

Various scenarios are considered. The most realistic is used in our study. In this scenario the evolution of traffic is based on extrapolation with a correction for improved vessel types, implementing more efficient loading and combined cargo.

5 PLACEMENT OF VESSELS IN THE LOCK

The placement process is controlled by an optimisation algorithm in such a way that the space in the lock chamber is filled in an optimal way. Further on we clarify the term 'optimal' in this type of application. The algorithm requires input data on the vessels, the locks and the control parameters of the algorithm. Per vessel, are required: an identification number, a length and a width. The units in which length and width are expressed are discussed later on. Information on the lock includes the following items: identification of name, length and width in the same units in which the size of the vessels are expressed. The control parameters include a length and width transformation parameter, a weight parameter and an exponent parameter.

Vessels are subdivided into *vessel types*. In the model a combination is made of five categories of vessels and six size classes for sea-going vessels. The categories include container ships, bulk carriers, RO-RO ships, tankers and general cargo vessels. The size classes are based on gross tonnage (in tons): <= 6000 (class 1), 6001-20000 (class 2), 20001-30000 (class 3), 30001-40000 (class 4), 40001-50000 (class 5) and > 50000 (class 6). For inland navigation four classes are defined: freight vessels, tankers, passenger vessels and leisure traffic, other non-freight carrying vessels.

When data are fitted to a distribution using the set of distributions available in the Arena software, a wide variety of distributions are found to be useful for fitting length, width and draught of the vessels. For the length of the vessels, the Beta, Triangular, Normal, Lognormal, Erlang and and Weibull distributions are chosen for various vessel type/size class combinations. In Table 1, some distributions with their parameter estimations are shown for the length of vessels of the type 'container vessel' (sea-going vessels).

Size class	Distribution type	Parameters
1	Beta	92 + 39 * BETA(0.752, 0.81)
2	Triangular	TRIA(117, 167, 201)
3	Triangular	TRIA(175, 210, 235)
4	Triangular	TRIA(187, 242, 265)
5	Beta	234 + 43 * BETA(1.25, 0.444)
6	Beta	270 + 25 * BETA(15.7, 2.25)

Table 1: Distribution parameters for the length of container vessels

In Table 2, similar distributions with their parameter estimations are shown for the width of vessels of the type 'container vessel'.

Size class	Distribution type	Parameters
1	Triangular	TRIA(13, 16.6, 21)
2	Triangular	TRIA(19, 27.6, 29)
3	Beta	25 + BETA(1.84, 1.22)
4	Weibull	30 + WEIB(2.21, 8.26)
5	Beta	32 + 0.351 * BETA(0.405, 0.628)
6	Beta	32.2 + 0.11 * BETA(0.592, 0.248)

Table 2: Distribution parameters for the width of container vessels

In Tables 1 and 2 the parameters between brackets for the various distributions have the following meanings. For the Beta-distribution (BETA) the values represent α_1 and α_2 so that its mean value = α_1 / (α_1 + α_2). For the triangular distribution (TRIA) the values represent the minimum value, the mode and the maximum value. For the Weibull distribution the values represent the shape parameter α and the scale parameter β so that its mean value = $\beta/\alpha\ \Gamma(1/\alpha)$ where $\Gamma(.)$ is the Gamma function (Pegden et al. 1995, Appendix A).

The placement of vessels in a lock resembles a classical OR problem, called the *packing* problem. It is an optimisation problem concerned with finding a best arrangement of multiple items in larger containing regions. The usual objective is to maximise material utilisation and hence to minimise the 'waste' area. In our case, the items are vessels. The containing region represents the lock. The waste area represents the surface of water not covered by vessels. Implicitly, we assume by minimising the waste area the same objective as putting a maximum number of vehicles in the lock. A review on packing problems can be found in Dowsland and Dowsland (1992).

The placement of vessels is a 2D-rectangular packing problem. A set of vessels has to be placed in a rectangular object, the lock. The lock is modelled as a rectangle with length L and width W where $L > W$. A vessel is also modelled as a rectangle with length l_v and width w_v where $l_v > w_v$. The placement process ensures there is no overlap between the items.

There are many variants of the 2D-rectangular packing problem. Dyckhoff (1990) has given a topology of packing problems in terms of the set of items, the region (single object), the rotation of items and the guillotine feature. Some problems require the items to be different, others allow for identical items. In our case, while not very probable, two identical vessels may approach the lock. The region, representing the lock in our reality, many times is assumed to be of fixed width and infinite height. However, the lock has finite height. The aspect of ***rotation*** relates to the possibility of rotating items by 90°. In the lock the vessels cannot be rotated and as the length is assumed to be larger than the width of the vessel, the larger part of the vessel is placed parallel to the length of the lock. Baker, Coffman and Rivest (1980) consider packings which ortogonal and oriented. An *orthogonal* packing is one in which every edge of every rectangle is parallel to iether the bottom edge or the vertical edges of the region. An orthogonal packing is also *oriented* if the rectangles are regarded strictly as ordered pairs; i.e. a rectangle (x_i, y_i) must be packed in such a way that the edges of length x_i are parallel to the bottom edge of the region. Thus rotation of 90° are not allowed. This feature is certainly applicable to our case.

The *guillotine* feature relates to a restriction of the partitioning of the lock area into rectangles. Guillotine cuts are cuts from one edge of a previously cut rectangle to the opposite edge. Cuts are assumed to be orthogonal. A cutting pattern restricted to those by guillotine cuts lends itself to the application of dynamic programming. The placement of vessels in a lock has not this type of patterns making this type of solution techniques not applicable.

The size of the search space of this placement problem is infinite, because every movement of the rectangle representing the vessel in to the placement pattern in a feasible direction creates a new pattern. The problem can be made *discrete* by assuming that lengths and widths are expressed in a basic unit. Units for expressing length and width are controlled by the user. The values of L and l_v are expressed in an elementary length unit Δx, and W and w_v are expressed in an elementary width unit Δy, which need not be equal to Δx. A vessel is placed in the lock with its length parallel to the length of the lock. Once a vessel is placed, it occupies an integer number of elementary rectangles of surface Δx times Δy.

The locks, which are of use in our real-world situation, are mentioned in Table 3. Their length, width and depth characteristics are shown. The length is only a fraction of the physical length as alternating one of both lock doors is die to maintenance. The depth is not the theoretical depth but the one, which is used in practice for lock planning.

Lock name	Length (in m)	Width (in m)	Depth (in m)
Zandvliet lock	473.0	57.0	12.1
Berendrecht lock	477.0	68.0	12.5
Van Cauwelaert lock	248.0	35.0	8.5
Boudewijn lock	337.0	45.0	9.0
Royers lock	180.0	22.0	4.5
Kallo lock	333.0	50.0	9.8

Table 3 : Lock dimensions leading to the Antwerp quays (in meter)

As the modelled length and width of both vessel and lock are integer multiples of their resp. elementary units, the length and width used in the algorithm are an approximation to the real length and width. The granularity is chosen by the user. But the user is warned that algorithm run times increase very fast with decreasing values of Δx and Δy.

Most packing studies start from an "open-ended" rectangle (or lock) which means it has a finite width but an infinite length. The optimisation problems, which are formulated, try to minimise the height, without considering the height is in fact a finite element. Within this idea, a class of heuristics has been formulated called the *bottom-up left-justified* (BL) algorithms (Baker, Coffman and Rivest 1980). Each such algorithm puts the vessels one at a time as they are drawn in sequence from the list of arriving vessels. When a vessel is placed in to the lock it is first placed into the lowest possible location, and then left-justified at this vertical location in the lock.

The type of orthogonal oriented problems mentioned earlier lead to *strip-packing* algorithms. Formally it can be defined as follows: Given a vertical strip of width 1, bounded below but not above, and a list L of rectangular pieces R_1, R_2, ... R_n , pack the pieces into the strip that the height to which the strip is filled is as small as possible. It is also assumed that each piece R_I in L is defined by its width a_i and its height b_i and must be packed in such a way that the edges corresponding to width (of length a_i) are parallel to the horizontal bottom edge of the strip. Strip packing can be extended to more than one strip, e.g. the following definition of the problem: "A variant of bin packing in which one must determine how to get the most objects from the least number of fixed length strips. More formally, find a partition and assignment of a set of objects of certain lengths such that they fit on a minimum number of strips" (http://www.nist.gov/dads/HTML/strippacking.html).

The problem of finding an optimal packing in this case is NP-hard, so approximation algorithms have been studied. Various approximation algorithms have been defined satisfying inequalities of the type:

$$A(L) <= \alpha \, OPT(L)$$

where A(L) denotes the height actually used by the algorithm, and

OPT(L) denote the minimum strip height within which the pieces from the list L can be packed. Examples of such approximation algorithms can be found in Golan (1981), in Sleator (1980) and in Steinberg (1997). Steinberg (1997) looks at a rectangle with finite height. He defines the conditions under which a set of rectangles can packed in a strip-packing way into the rectangle with finite height. But the conditions assume that the rectangles can be packed in any sequence which is not true in our case.

A special class of algorithms for the packing problem are the *on-line* algorithms. An algorithm is called an on-line algorithm if it packs the items in the order given by a list without knowledge of the subsequent items on the list. When each rectangle arrives it must be immediately assigned its place in the bin. Only once this has happened, does the identity of the next rectangle become known. In our real-world application this feature is only partially true. It is not really true that we are not aware of the arrivals of any subsequent vessels. While vessels arrive at the lock, the planners have knowledge about arriving vessels from the river mouth, which more than 100 km farther on. But it is true, in some sense, as the sequence of arrivals is the same as the sequence of placements. Vessels do not wait in front of the lock to allow the planners to make a better utilisation of the lock. In this family of applications, Lee and Lee (1985) introduced the so-called Harmonic algorithm for on-line bin packing. A lower bound for on-line bin packing is obtained by van Vliet (1992). Baker and Schwarz (1983) mention a situation similar to ours, but for general packing and not strip-packing. They consider the rectangles to form a queue, which can be joined by new rectangles at any time. Sometimes in such situations it is not reasonable to wait for all rectangles to enter the queue before packing them. They define a class of heuristics, called *shelf* algorithms, which pack the rectangles in the order specified by the queue.

If all these ideas are joined into the strip packing problems, then five approaches to heuristics can be developed (www.cs.cf.ac.uk/user/C.L.Valenzuela/heidi/Algorithms.html): bottom left, level-oriented, split, shelf or hybrid. The *Bottom-up Left-justified* algorithm requires each successive vessel to be placed as near as the outgoing gate of the lock as it will fit and then as far as left as it can go at that level without overlapping any other already placed vehicles. No pre-sorting is required, so this can apply to our queue-like situation In the *Level-oriented* algorithms the packing is done on a series of levels that the bottom of each rectangle rests on. As the vehicles need to be sorted into order of decreasing height, this is of no relevance to our case. *Split* algorithms split the open-ended bin vertically into smaller open-ended bins depending on the widths of the rectangles. Also here some pre-sorting is required, in this case sorted by width. This means it is also of no use to us. *Shelf* algorithms are modifications of the level algorithms that avoid pre-sorting. The levels are fixed height shelves. The shelf heights are set by a parameter. Once all appropriate shelves are defined such a shelf algorithm needs a slave algorithm to decide which shelf a rectangle should be put in. Baker and Schwarz, for example, define the First Fits Shelf (FFS)-algorithms as a slave. A new rectangle is packed into the lowermost shelf (read in out case, the area closes to the outgoing gate) into it will fit. While shelves are not required in our case, this heuristic could give a first solution from which further improvements could be built. *Hybrid* algorithms use characteristics of several types of the previously described heuristics.

Depending on the time available for the lock planners, a simple and quick approach should be used or some time could be spent on some improvement. In the following we illustrate some ideas how further improvement could be done, by making use of a special objective function.

A *section* on x is a cut over the whole width, parallel to the width axis of the lock, made at a point x on the length axis. The set of points where a cut can be made is limited to the points 0, Δx, $2\Delta x$, $3\Delta x$, .., $n\Delta x = L$.

To determine an optimal location of the vessels, the algorithm requires an objective function. The function value can be determined at any point where a section is allowed. Let $V(0,x)$ be the objective function value of the planning for the rectangle defined by the gate location (point 0) up to the section at distance x.

In our case the objective function takes two elements into consideration: (1) how well is the lock occupied by a number of vessels to which a location is assigned (denoted as *OS(0,x)*) ; and (2) how well can the remaining space be used (denoted as *RS(0,x)*).

An elementary rectangle has an attribute called *state*. The state can take three values: 'Free', 'Occupied' or 'Unreachable'. A rectangle with state 'Unreachable' has value zero and does not contribute to the objective function. The rectangle moves into this state if the space is enclosed and, by this, is of no use for further allocations of vessels. For example, if the rectangle $[(x_i,y_i),(x_{i+1},y_i)]$ is not yet occupied, but the rectangle $[(x_{i+1},y_i),(x_{i+2},y_i)]$ is occupied, the former rectangle is unreachable. The objective function, defined at a point x, is obtained by combining the values of the rectangles having state 'Occupied', *OS(0,x)* with those having state 'Free', *RS(0,x)* or

$$V(0,x) = \mathbf{f}\,(OS\,(0,x)\,,\,RS(0,x)\,).$$

Both the functional form **f** and the way of calculating *OS (0,x)* and *RS(0,x)* are not at all random. It seems plausible the value of a full lock occupation *V(0,L)* be equal to W times L. An optimal location, not leading to a full occupation of the lock, must have a value lower than W * L. The function for *OS (0,x)* can be of additive nature. If a first vessel is placed in the area $[(x_0,y_0),(x_2,y_0)]$ and a second one in the area $[(x_0,y_2),(x_2,y_2)]$, this is considered to be equal to an occupation in which the vessels are located in the areas $[(x_0,y_0),(x_2,y_0)]$ and $[(x_0,y_1),(x_2,y_1)]$.

In terms of remaining surface the second situation is of more future use. Both values of $OS(0,x_2)$ are equal while the value of van $RS(0,x_2)$ is higher in the second situation. It is clear that e.g. three rectangles of elementary surface offer less future opportunities than a strip of three contiguous elementary rectangles.

A linear additive function is not feasible for the value of *RS(0,x)*. A power function has the required qualitative properties but needs an additional parameter input from the user. This parameter is one of the control parameters and is called the *exponent.* In terms of remaining surface there exists an interaction between horizontal and vertical strips. The value shall be determined on basis of counting horizontal and vertical rectangles. The interaction requires another parameter called the *weight.*

The allocation of vessels is done via a heuristic based on dynamic programming. This technique has a recursive character so for implementation purposes the use of dynamic data structures as lists are appropriate. The technique of dynamic programming in its search for the optimal solution is limited in practical applications due to the large number of intermediate solutions. Practical implementations will need to make use of sub-optimal solutions in which both computing time and storage are limited.

For performance reasons it is recommended that not an optimal evaluation is searched for a set of vessels which potentially can enter the lock. It is much more efficient that for a representative distribution of vessel types an optimal parameter set is computed off-line. This set is used during the simulation program so that no extra time is lost. We are aware that this is not the best solution in terms of scheduling with known vessel identifications and characteristics, because these parameters are considered 'optimal' on the average or representative set of vessels. For practical purposes, this optimisation within the simulation model is an improvement over common practice.

6 EMBEDDING THE PACKING MODEL INTO A SIMULATION MODEL

The planning of the lock operations is based on available data. A trade-off is made between two situations: better utilisation of the lock or faster throughput of the vessels. For incoming traffic, the data are known from the moment the vessels enter the river Scheldt either at one of the following two locations: A1-buoys/Wandelaar or Steenbank. The information in sequence of arrival at one of those entry-points allows the planners to make a planning but this planning is not easy due to various factors. They include: some vessels can overtake other vessels, vessels move at different speed, some vessels are tide-dependent, priority rules exist among sea-going vessels and inland navigation.

Independent of the number of vessels in the lock, each of them is confronted with a delay called the lockage time. The locks are used because there is a tide on the river while the water levels in the docks in the port is nearly constant. The gates can only open when the level in the lock chamber is equal to the water level upstream of the upper gates or downstream of the lower gates. The filling and emptying of the lock chamber is accomplished by opening and closing two sets of gated valves which control the flow of water into or out of the culverts in the lock walls. The water level continues to raise or fall until it is equal to the pool level upstream or downstream of the gate. The gates may then be opened to allow the vessels to enter of exit.

In the simulation study the lockage times have been considered to be random variables. From real-life data, distributions were estimated for the various lock on the river Scheldt. The data are shown in Table 4.

Lock name	**Distribution type**	**Parameter estimation**
Zandvliet lock	Beta	10.5 + 29.5 * BETA(8.82, 10.2)
Berendrecht lock	Poisson	POIS(24.3)
Van Cauwelaert lock	Lognormal	5 + LOGN(15.9, 3.06)
Boudewijn lock	Lognormal	10 + LOGN(13.3, 2.81)
Royers lock	Gamma	5 + GAMM(5.35, 3.37)
Kallo lock	Normal	NORM(14.8, 3.65)

Table 4 : Lockage time of the river Scheldt locks

The arrivals of vessels in the river have to be translated in the model into arrivals for the locks. Vessels move to one of the locks to enter the port but the outgoing lock may be a different one. We illustrate this by means of one of the classes of inland waterway traffic, the freight vessels, in Table 5.

	SRC	**ZABE**	**BOVC**	**RO**	**KA**	**AC**	**AQ**	**KT**
Incoming	46%	4%	7%	12%	4%	23%	1%	3%
Outgoing	42%	4%	7%	14%	4%	26%	1%	2%

Table 5 : Percentage distribution of lock destination for inland waterway freight vessels

In Table 5 the abbreviations have the meanings: Scheldt-Rhine connection (SRC), Lock complex Zandvliet-Berendrecht (ZABE), Lock complex Boudewijn-Van Cauwelaert (BOVC), Royers lock (RO), Kallo lock (KA), Albert channel (AC), Antwerp quays (AQ), Kattendijk lock (KT).

The arrivals at the Steenbank or A1 entry points on the river have to be corrected for some type of vessels, which are called type-dependent vessels. As the river has many places which are not very deep, vessels with a large draught can sail only during a limited interval of the tide period. This period is called the tide window. Based on the physical data of the vessels and the moment of arrival, a decision can be made whether the vessel can sail directly to its destination lock or it has to wait one of two potential shelters on the river. By this effect also the ordering of vessels can de changed. Also vessels sail at different speeds, e.g. container vessels sail at 15 to 16 knots, while inland waterway freight vessels sail only at 5.8 till 10.8 knots depending on their size class.

7 CONCLUSION

A reusable river port simulation model is extended with locks giving access to the port of Antwerp. A strategy to fill the locks to the highest possible capacity requires the use of an optimisation algorithm. The fundamentals of a heuristic algorithm the placement of vessels within the locks is described. The optimisation algorithm is embedded into the simulation model, which has been implemented in the Arena simulation software. The data were taken from a study by the consultancy company Orinoco (1997).

REFERENCES

Baker B.S., E.G. Coffman and R.L. Rivest 1980. Orthogonal packing in two dimensions, *SIAM Journal on Computing*, vol. 9 no. 4, pp. 846-855.

Baker B.S. and J.S. Schwarz. 1983. Shelf algorithms for two-dimensional packing problems, *SIAM Journal on Computing*, vol. 12 no. 3, pp. 508-517.

Bandy D.B. 1996. Simulation of the Illinois waterway locks system. In: Charnes J.M., D.J. Morrice, D.T. Brunner and J.J. Swain (eds.), *Proc. 1996 Winter Simulation Conference* (Coronado, CA), pp. 1483-1487.

Blümel E. (ed.) 1997. *Managing and Controlling Growing Harbour Terminals*. Results of the Copernicus 1994 Project 312 AMCAI, Society for Computer Simulation, San Diego/ Erlangen/ Ghent/ Budapest.

Blümel E. and L. Novitsky (eds.) 2000. *Simulation and Information Systems Design: Applications in Latvian Ports.*Results of the Copernicus 1998 Project PL976012 DAMAC-HP (ISBN 9984-606-73-2), Magdeburg.

Bolten J.G. 1980. *Inland shipping and water management decisions in the Netherlands.* Ph.D. Thesis, The Rand Graduate Institute, Santa Monica, California.

Coeck C., T. Notteboom, A. Verbeke and W. Winkelmans, 1996, Back to the future: an evaluation of traffic forecasting techniques in Antwerp and Rotterdam. In: Winkelmans W. and T. Notteboom (eds.), *The Issue of Prognoses in Maritime Transport and Seaport Production in Belgium and Poland: a Conceptual and Empirical Study,* Garant, Leuven/Apeldoorn, pp. 3-26.

Dowsland K.A. and W.B. Dowsland. 1992. Packing problems, *European Journal of Operational Research,* vol. 56, pp. 2-14.

Dyckhoff H. 1990. A typology of cutting and packing problems, *European Journal of Operational Research,* vol. 44, pp. 145-159.

Fleming D.K. 1997. World container port ranking, *Maritime Policy and Management,* vol. 24, no. 2, pp. 175-181.

Golan I. 1981. Performance bounds for orthogonal oriented two-dimensional packing algorithms, *SIAM Journal on Computing,* vol. 10, pp. 571-582.

Hassan S.A. 1993. Port activity simulation: an overview, *Simulation Digest,* vol. 23 no. 2, pp. 17-36.

Hayuth Y., M.A. Pollatschek and Y. Roll. 1994. Building a port simulator, *Simulation,* vol. 63, pp. 179-189.

Janssens G.K., G.F. Thiers and W. Tielemans, 1999. Polyvalent pilotage in the Flanders coastal and inland ports, In: Bruzzone A.G., Y.A. Merkuryev and R. Mosca (eds.), *Proc. of the International Workshop on Harbour, Maritime & Industrial Logistics Modelling and Simulation, September 1999, Genoa, Italy,* pp. 181-188.

Kondratowicz L. 1994. *Computer simulation in logistical analysis of multimodal transport: methodology and implementation,* Publications from the Center for Maritime Studies, University of Turku, no. A15, Turku, Finland.

Kooman C. and P.A. de Bruijn. 1975. *Lock capacity and traffic resistance of locks.* Rijkswaterstaat Communications no. 22, Government Publishing Office, The Hague.

Kuehn, W., M. Tabucanon, H. Pastijn, E. Oey and S. Thinapong. 1994. Simulation approach for performance evaluation of an industrial port system. In: Kuehn W. and N.N. Nagarur (eds.), *Simulation in Manufacturing Systems: Proc. 1st International Workshop WORKSIMS '94,* November 1994, Bangkok, Thailand, pp. 228-234.

Lee C.C. and D.T. Lee. 1985. A simple on-line bin packing algorithm, *Journal of the ACM,* vol. 32, pp. 562-572.

Martinelli D., M.D. Dai, P. Schonfeld and G. Antle. 1993. Methodology for planning investments on inland waterways. *Transportation Research Board,* no. 1383, pp. 49-57.

Orinoco n.v. 1997, Simulatiestudie van de maritieme verkeersafwikkling in de Antwerpse haven en op de Westerschelde tengevolge van de buitendienststelling en renovatie van de Royerssluis (in Dutch, Simulation study of maritime traffic assignment in the port of Antwerp and on the Western Scheldt river as a result of the unavailability and renovation of the Royers lock).

Pegden C.D., R.E. Shannon and R.P. Sadowski. 1995. *Introduction to Simulation using SIMAN,* McGraw-Hill, New York, 600 pp.

Sleator D. 1980. A 2.5 times optimal algorithm for packing in two dimensions, *Information Processing Letters,* vol. 10, pp. 37-40.

Steinberg A. 1997. A strip-packing algorithm with absolute bound 2, *SIAM Journal on Computing,* vol. 26, no. 2, pp. 401-409.

Taeymans B. and M. Bouwman. 1998. A vessel traffic simulation model for the North Sea Channel Area: development of an advanced visual component library for building vessel traffic simulation models. In: *PORTS 98 Conference, (*Longbeach CA).

Thiers G.F. and G.K. Janssens. 1998. A port simulation model as a permanent decision instrument, *Simulation* , vol. 71 no. 2, pp. 117-125.

Toluyev J., Y. Merkuryev, E. Blümel and M. Nikitins. 1996. *Port terminal simulation: a survey,* Riga Technical University, Latvia.

Van Vliet A. 1992. An improved lower bound for on-line bin packing algorithms, *Information Processing Letters,* vol. 43, pp. 277-284.

Wadhwa L.C. 1992. Planning operations of bulk loading terminals by simulation, *Journal of Waterways, Port, Coastal and Ocean Engineering,* vol. 118 no. 3, pp. 300-315.