Decision support system for container terminal planning

Citation for published version (APA):

Document status and date:
Published: 01/01/1986

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
DECISION SUPPORT SYSTEM FOR
CONTAINER TERMINAL PLANNING

by

K.M. van Hee and R.J. Wijbrands
DECISION SUPPORT SYSTEM FOR CONTAINER TERMINAL PLANNING

by

K.M. van Hee and R.J. Wijbrands

1. Introduction

The decision support system (dss) we will describe is developed for capacity planning of container terminals. The system is originally built to support the design process of a new terminal, however, it can be used also for tuning existing terminals.

The dss incorporates several mathematical models, each describing parts of the process of container handling on a terminal. The relationships between these models is one of the interesting features of the dss. There is a hierarchy in the set of models. Models which consider the process in more detail are used to compute parameters of models that consider the process on a more global level. We distinguish three levels (see section 2 for a more detailed description of the terminal process):

a. Model for the behaviour of one gantry crane used to stack containers on the yard (called: yard stackers). With this model characteristics of the service time distribution of these cranes are computed. They are used in b.

b. Model for the interaction of terminal equipment such as: yard stackers, slave trailers and gantry cranes for ship loading (called: ship cranes) in a period of time (hours) where the number of ships at the quay and the arrival rate of external trucks is constant. With this model the throughputs of the ship cranes are computed. These quantities are used in c.

c. Models for the behaviour of the whole terminal in a longer period of time (days, weeks or months). This model computes the external behavioural characteristics of the terminal such as quay occupation, berth time distribution and lost calls.
Models of c are similar to some of the models we described in [van Hee, Huitink and Leegwater (1985)].
The dss can be used to support several types of decisions:

1. **strategic decisions** such as quay capacity, yard layout and type of cranes;
2. **tactical decisions** such as the choice of the rules for the operational managers to allocate containers on the yard and to extend or reduce the number of trailers for the transport of containers from or to a ship.

In section 2 we sketch the process of a container terminal. In section 3 we consider the model for the yard stacker (model a). In section 4 we describe model b. In section 5 we make some remarks on models c and in section 6 we sketch the architecture of the dss.

Our models are all analytical, i.e. we do not use simulation models in the dss. The reason why is that simulation models for the process of a container terminal are very time consuming and therefore difficult to use in an interactive way. However, the decision maker wants to vary parameters very frequently to determine his decision and therefore he needs interaction.

### 2. Container terminal process

Container terminals form an essential link in the transport chain of a container. It fulfils two functions: the transshipment and the temporary storage of containers. Container carriers are large compared to the more classical breakbulk ships. They can carry a few thousand containers. They are expensive so the berth times of these ships must be as small as possible. Container terminals can be organized in different ways with different types of equipment. We consider a rather familiar type of organization, however most of the methods we describe can be used for other organization forms as well.
In figure 1 we sketch a terminal:

A: shipcrane  
B: yard stacker  
C: slave trailers  
D: external trucks

External trucks bring export containers to the terminal and (or) take import containers from the terminal. These trucks go to a yard stacker that picks the export container from the truck and puts it into the stack yard, usually into a part that contains only containers with the same destination. A truck that picks up an import container moves to the part of the stack yard where the import containers are placed and waits for a yard stacker. The yard stackers also serve the internal trucks called slave trailers that bring containers from or to the one shipcrane. They may go to several yard stackers, but return to the same shipcrane.

The stack yard is usually divided into compartments: per ship for import and export, whereas the compartments in the export stack are further divided into compartments per destination. The determination of the size and the location of each compartment is an important operational decision. Export containers that are delivered long before the departure of the ship are sometimes stored in a buffer compartment of the stack yard. They are brought to their compartment as soon as it is allocated.

To load the export containers as quick as possible it is important that containers are placed in the yard in the reversed order of their positions in the ship (see figure 2):
In case of yard stackers the columns may be interchanged, however, if fork lift trucks are used instead of yard stackers, the columns must be as indicated.

The external trucks arrive at random at the terminal. This implies that there is a positive chance that the container they will pick up is not on top of one of the stacks. Therefore the yard stacker has to reshuffle the stack in which the container is. This takes time.

On the other hand, the export containers might be delivered in a wrong order. This last problem can be solved by letting the yard stackers do the reshuffling of export compartments at moments they are idle. Usually all export containers have to be delivered some time before their ship arrives, so there is some time left. Therefore we do not pay much attention to the reshuffling of export containers in the model. For the import containers this reshuffling has to be done at 'real-time' and therefore we must take this into account in the models.

At first sight one may think that reshuffling of import containers in the idle periods of yard stackers is also useful, namely ordering the containers in one stack according to their residence time: 'old on top of young'. However, statistical investigations have shown that younger containers have a higher chance to be picked up at time t than older ones, given that they are both at terminal at time t. (In fact, there is a positive chance that a container is never collected.)

The performance of a terminal can be expressed in the following quantities:

- Number of lost calls (unserved ships).
  It is assumed that ships go to an other terminal if there is no berth. (This happens seldom.)
- Average ship delay (compared to the minimal service time).
- Waiting time of inland transports (here we only consider trucks).
The characteristics of the efficiency of the terminal are:

- utilization of human resources and equipment;
- yard utilization;
- quay utilization;
- cost of the operation.

Human resources are not considered separately because they are coupled to the equipment.

The relationships between the models indicated in section 1 becomes clear now. The performance of the yard stacker is influenced by the intensity of the external trucks and the division of its service over import and export, but also by the number of slave trailers it has to serve. The net service time distribution will depend on these influences. The service time distribution itself influences the cycle times of the slave trailers and therefore the throughput of the shipcranes. This quantity determines the gross service time of the ships.

The arrival pattern of external trucks may be modelled by an inhomogeneous Poisson process, with an arrival rate that varies over the day. In the models we assume the arrival rate to be constant in periods of one hour. Hence the throughput of the shipcranes varies from hour to hour.

3. The yard stacker

At the most detailed level of our container terminal model, we study each of the model elements as stand alone systems. In this way we can investigate the influence of traffic intensities and system utilization on the service rate of each of these individual elements. As an example we shall discuss a model for a yard stacker serving the import stack yard. Other traffic sensitive model elements, such as the yard stackers serving the export stack yard or the vessels, can be dealt with in a similar way.

The yard stacker which handles the containers in the import stack yard is a huge moveable crane. It can move in the length (bridge movement) as well as in the width (trolley movement) of the stack yard. Two types of traffic arrive at the yard stacker. First we have the external trucks, which come
to pick up containers for further inland transportation. Secondly we have the so-called slave trailers which deliver the containers they just obtained from a ship for temporary storage. For practical reasons, the organization at the stack yard has been chosen such that the external trucks and slave trailers are separated by the yard (see figure 1). As a consequence of this the yard stacker repeatedly has to cross the yard; alternately the slave trailers and the external trucks have to be served. Several service order disciplines could be applied here. For the moment we assume that the service order is first come first served. The alternating service pattern causes the time for handling a container to depend on the preceding operation. If the current operation is a delivery by a slave trailer, whereas the preceding operation was a request from an external truck, then the current operation is starting with the trolley crossing the yard. If the preceding operation was a delivery as well, then the trolley movement would have been limited; assuming a random start position only half the stack yard had to be crossed. This dependency between two successive operations is largely determined by the arrival intensities of the two types of traffic.

At this level of the container terminal model, the arrival intensities are assumed to be given. An estimate of the arrival rate of the slave trailers can be obtained by iteratively using the global container terminal model, thus computing the throughput of the slave trailers, and the detailed model of the yard stacker which yields, using the updated knowledge about the arrival intensities, new estimates for the average service times.

We may partition the operation for handling a container delivered by a slave trailer or requested by a truck in a number of steps. In figure 3 this partitioning is given for each of the possible start positions. For each of the steps we can estimate the average (and standard deviation) of its service time by making some simple assumptions. For all movements, of the bridge, the trolley and the hook to which the containers are to be attached, we assume a constant acceleration until a maximum speed is reached. Thus travelling a distance $d$ with acceleration/ deceleration $a$ and a maximum speed $s$ requires a time $T(d)$:
<table>
<thead>
<tr>
<th>Current operation</th>
<th>Preceding operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Putting a container into the stack yard</td>
<td>Putting a container into the stack yard</td>
</tr>
<tr>
<td>I START: hook high above the stack yard</td>
<td>III START: hook high above the stack yard</td>
</tr>
<tr>
<td>(a) drive to the slave; trolley movement</td>
<td>(a) drive to the requested container; both bridge and trolley movement</td>
</tr>
<tr>
<td>(b) hook down (uncharged)</td>
<td>(b) hook down (uncharged)</td>
</tr>
<tr>
<td>(c) attach container</td>
<td>(c) (remove, if needed, the containers which are stored above the requested container)</td>
</tr>
<tr>
<td>(d) hook up (charged)</td>
<td>(d) attach container</td>
</tr>
<tr>
<td>(e) drive to the proper position, trolley movement</td>
<td>(e) hook up (charged)</td>
</tr>
<tr>
<td>(f) hook down (charged)</td>
<td>(f) drive to the truck; trolley movement</td>
</tr>
<tr>
<td>(g) detach container</td>
<td>(g) hook down (charged)</td>
</tr>
<tr>
<td>(h) hook up (uncharged)</td>
<td>(h) detach container</td>
</tr>
<tr>
<td>STOP</td>
<td>(i) hook up (uncharged)</td>
</tr>
<tr>
<td>STOP</td>
<td>STOP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Delivering a container from the stack yard</th>
<th>Delivering a container from the stack yard</th>
</tr>
</thead>
<tbody>
<tr>
<td>II START: hook up at the side of the stack yard</td>
<td>IV START: hook up at the side of the stack yard</td>
</tr>
<tr>
<td>(a) cross the stack yard; trolley movement</td>
<td>(a) drive to the requested container; both bridge and trolley movement</td>
</tr>
<tr>
<td>(b) hook down (uncharged)</td>
<td>(b) hook down (uncharged)</td>
</tr>
<tr>
<td>(c) attach container</td>
<td>(c) (remove, if needed, the containers which are stored above the requested container)</td>
</tr>
<tr>
<td>(d) hook up (charged)</td>
<td>(d) attach container</td>
</tr>
<tr>
<td>(e) drive to the proper position; trolley movement</td>
<td>(e) hook up (charged)</td>
</tr>
<tr>
<td>(f) hook down (charged)</td>
<td>(f) drive to the truck; trolley movement</td>
</tr>
<tr>
<td>(g) detach container</td>
<td>(g) hook down (charged)</td>
</tr>
<tr>
<td>(h) hook up (uncharged)</td>
<td>(h) detach container</td>
</tr>
<tr>
<td>STOP</td>
<td>(i) hook up (uncharged)</td>
</tr>
<tr>
<td>STOP</td>
<td>STOP</td>
</tr>
</tbody>
</table>

Figure 3. Detailed reconstruction of the operations performed by a yard stacker at the import stack yard.
The dimensions of the stack yard added to the acceleration and speed of the yard stacker immediately determine nine of the operations listed in figure 3: I b and d, II a, b and d, III g and i and IV g and i. Note that we assumed that the slave trailers and trucks are always positioned at the same field width position as the yard stacker. This to reduce the bridge movement. Furthermore it is assumed that a container delivered by a slave trailer can always be stored at one of the stacks in the current field width position. Both assumptions are based on experiences in the practice. Although the positioning of the containers in the stack yard is one of the major concerns of the yard manager, we here assume that the containers are stored at random over the stack yard. We can make this assumption as the containers arrive and depart from the stack yard in a more or less random order. If \( w \) denotes the width of the stack yard, the time required for the trolley to move from a random position to the side of the stack yard is determined through

\[
\int_0^w \frac{1}{w} T(x) dx = \frac{1}{2} \frac{w}{s} + \frac{s^3}{6} \frac{1}{wa^2}.
\]

The time required for moving from a random position to another random position is determined by

\[
\int_0^w \frac{2}{w} (1 - \frac{x}{w}) T(x) dx = \frac{1}{3} \frac{w}{s} + \frac{s^3}{3} \frac{1}{wa^2} + \frac{13}{15} \frac{s^5}{2wa^3}.
\]

In both equations we assumed the distance to be large enough to allow the maximum speed.

Noting that the bridge and trolley movements may take place at the same time, we have determined the operations I a en e, II e, III a en f, and IV a en f.

The travel distance distribution for the hook of the yard stacker, picking a container from, or putting a container in the yard, is determined by the distribution of the stack height. It is easily verified that this
distribution, given the random placement of the containers, is the hyper-
geometrical distribution. Now let us define \( h \) the maximum height of stacking
containers, \( M \) the actual number of containers in the stack yard and \( N \) the
maximum number of containers in the stack yard. Then, the probability that
a stack contains \( i \) empty places, given the utilization of the stack yard
is given through:

\[
H(i | M) = \frac{\binom{h}{i} \binom{N-h}{N-M-i}}{\binom{N}{N-M}}.
\]

The conditional probability that a delivered container is placed on top of
a stack with \( i \) empty places, again given \( M \), is proportional to the number
of empty places in that stack:

\[
\mathbb{P}[i | M] = \frac{i H(i | M)}{\sum_{j=0}^{h-1} j H(j | M)} = \frac{i}{N-M} H(i | M) \frac{N}{h}.
\]

It can be shown that to calculate the unconditional probability only the
first \( h-1 \) moments of the distribution of \( M \) are required. It is our
experience that the type of distribution of \( M \) is not of much influence to
the final results. In our models we have assumed that \( M \) is determined by a
normal distribution.

With this probability distribution we can compute the hook travel times \( I \)
\( f \) and \( h \) and \( II \) \( f \) and \( h \). In an analogous way we can compute the time
required for lifting the top container of a stack \( III \) \( g \) and \( i \) and \( IV \) \( g \) and
\( i \). Note that the average number of free places in the stack is in this
case approximately one container less than in the case we add a container
\( (N \text{ big, } q \text{ utilization of the stack yard } M/N) \)

\[
\sum_{i=0}^{h} \frac{(h-i)}{M} H(i | M) \frac{N}{h} \approx h(1-q) + q - 1
\]

\[
\approx \sum_{i=0}^{h} \frac{i^2}{N-M} H(i | M) \frac{N}{h} - 1
\]

If we assume the line required for attaching and detaching container and
hook to be approximately a constant, the only operation left which
requires some explanation in figure 3 is the shifting of containers to get
access to a container not on top of the stack. To compute the hook travel distance, we have already found the distribution of the stack height. As we have requested a random container this gives us an idea about the number of containers to be shifted as well. Shifting a container requires several small movements of trolley, bridge and hook. Knowing the stack height distribution we can compute the time required to move a container to the closest possible position.

Now we are able to compute the average service time for handling a container. If the arrival rate of slave trailers is \( \lambda_1 \) and the arrival rate of trucks is \( \lambda_2 \) we find

\[
\frac{\lambda_1}{\lambda_1 + \lambda_2} (I+III) + \frac{\lambda_2}{\lambda_1 + \lambda_2} (II+IV)
\]

For each step in the set of operations required to handle a container we have defined a small model depending on a number of parameters such as stack length, width and height, and yard stacker speed. The inclusion of these parameters into our model allows us to use the model in the performance optimization of the container terminal under study. In figure 4 we give a graphic representation of the optimal cycle time (average service time) of the yard stacker at the import stack yards. As a constraint in this optimization problem the number of containers in the stack was fixed. All velocities were assumed to be constant. Further data we used:

- maximum stack height in containers: 6
- speed (in containers):
  - of trolley: 32.7
  - of bridge: 9.6
  - of hook (charged): 11.5
  - of hook (uncharged): 23.1
- time for attaching/detaching containers and hook (min.): 0.25
- average filling level: 0.59
- capacity of territory (in containers): 870

Trucks and slaves arrive with equal intensities.

In figure 5 the solution for the same problem is given, but with the restriction added that the length of the stack yard is 16 containers at most. Note that the optimal stack height has shifted from two to three.
Figure 4. Fixed number of containers to be served, unrestricted length of service territory.

Figure 5. The same situation as in figure 4, however the length of the service territory of the crane is restricted to $20 \times 40$ foot.
4. Model for the interaction of equipment

In this model we assume the service time of the yard stacker to be known. The interaction of yard stackers, slave trailers and shipcranes is modelled by a separable queuing network (cf. [Baskett, Chandy, Muntz & Palacios (1975)]. Each shipcrane and each yard stacker is considered to be a server in the network. Each server is assumed to have an exponentially distributed service time. Closed and semi-closed queuing networks appear to be quite robust for this assumption, i.e. if the service distributions are not exponential, the results in the form of throughputs are not much different from the case where they are exponential, provided they both have the same mean value.

The customers in the queuing network are of two types: the slave trailers and the external trucks. The slave trailers remain in the system and they form in fact a closed network. The external trucks are only visiting one server and disappear afterwards. They arrive according to a Poisson process. Each of the servers can handle one customer at a time. The travelling time from the yard stacker to the shipcrane and vice versa is modelled by another server with an infinite service capacity. (Hence the travel time may be nondeterministic.)

The quantities we want to compute for these networks are all mean values. Therefore we may use the solution technique developed by Reiser and Lavenberg and extended by Zahorjan and Wong, called mean-value analysis. The results of mean-value analysis hold for stationary situations only. However, it turns out that the networks we consider stabilize very fast: after a few cycles of slave trailers.

The parameters are:

- \( I \) = set of shipcranes
- \( J \) = set of yard stackers
- \( p(i,j) \) = probability that a slave trailer that belongs to shipcrane \( i \) will go to yard stacker \( j \)  
  (Note that \( \sum_{j \in J} p(i,j) = 1 \).)
- \( d(i,j) \) = mean travel time from shipcrane \( i \) to yard stacker \( j \)  
  (Note that we do not require \( d(i,j) = d(j,i) \).)
- \( \lambda(j) \) = mean arrival rate of external trucks at yard stacker \( j \)
ss(i) = mean service time at shipcrane i
sy(j) = mean service time at yard stacker j
k(i) = number of slave trailers assigned to shipcrane i
K = the vector with #(I) components k(i)
e_i = vector with #(I) components all of which are zero except for the i-th which is one.

Note that sy(j) is computed with the yard stacker model.
Further we define variables:

- \( \lambda_i(K) \) = mean number of slave trailers at shipcrane i
- \( m_j(K) \) = mean number of slave trailers at yard stacker j
- \( n_j(K) \) = mean number of external trucks at yard stacker j
- \( v_i(K) \) = mean residence time of a slave trailer at shipcrane i
- \( w_j(K) \) = mean residence time of a slave trailer at yard stacker j
- \( z_j(K) \) = mean residence time of an external truck at yard stacker j
- \( t_i(K) \) = throughput of shipcrane i, i.e. the mean number of slave trailers that leaves the shipcrane per time unit.

In the algorithm we use the well-known formula of Little and the formula for the mean queue length in an M/M/1 queue (cf. [Kleinrock (1975)]). We first give the algorithm and then we give an explanation.

**Algorithm**

a. Initialization:

\[ K := (0, \ldots, 0); \]

For all \( j \in J \):
\[ m_j(K) := 0; \lambda_j(K) := 0; \]
\[ n_j(K) := \lambda(j)sy(j) / (1-\lambda(j)sy(j)) \]

(1)

b. Recursion over K:

For all \( i \in I \):
\[ v_i(K) := ss(i)(1+\lambda_i(K-e_i)) \]

For all \( j \in J \):
\[ w_j(K) := sy(j)(1+m_j(K-e_j)+n_j(K-e_j)) \]

(3)
For all $i \in I$:

\begin{equation}
\tag{4}
t_i(K) := k(i) / \left( v_i(K) + \sum_{j \in J} p(i,j)(w_j(K) + d(i,j) + d(j,i)) \right)
\end{equation}

\begin{equation}
\tag{5}
\xi_i(K) := t_i(K)v_i(K)
\end{equation}

For all $j \in J$:

\begin{equation}
\tag{6}
m_j(K) := w_j(K) \sum_{i \in I} p(i,j)t_i(K)
\end{equation}

\begin{equation}
\tag{7}
n_j(K) := \lambda(j) s_y(j)(1 + m_j(K) + n_j(K)) / (1 - \lambda(j) s_y(j))
\end{equation}

Note that (1) is the mean queue length in an $M/M/1$ queuing system. This formula is intuitively correct since there are no slave trailers in this situation. Further, (5) is Little's formula. Formula (7) is the solution of a pair of equations

\[ z_j(K) = s_y(j)(1 + m_j(K) + n_j(K)) \]

and

\[ n_j(K) = \lambda(j) z_j(K). \]

Note that in mean-value analysis a closed-loop customer arriving at a server finds the server in an equilibrium state of the process with one customer of his own type less. An open-loop customer, the external trucks, finds the process in an equilibrium state with all closed and open loop customers. (This last property is referred to as PASTA: Poisson arrivals see time averages (cf. [Wolff (1982)]).)

The computed characteristics of the efficiency are:

- throughput of shipcrane $i$: $t_i(K)$
- utilization of shipcrane $i$: $t_i(K)s_s(i)$
- utilization of yard stacker $j$: $s_y(j)\left( \sum_{i \in I} p(i,j)t_i(K) + \lambda(j) \right)$
- utilization of slave trailers connected to shipcrane $i$:
  \[ 1 - \left\{ v_i(K) - s_s(i) + \sum_{j \in J} p(i,j)(w_j(K) - s_y(j)) \right\} t_i(K) / k(i) \]
- mean waiting time of external trucks:
  \[ \sum_{j \in J} \lambda(j)w_j(K) / \sum_{j \in J} \lambda(j). \]
In fact we are not only interested in the mean waiting time but also in an 'upperbound'. We define this as the point where the excess probability is smaller than or equal to \( \alpha \).

We cannot compute this value exactly, however we give an approximation based on the following heuristic:

- Model the process of external trucks at the yard stacker \( j \) by an \( M/M/1 \) queuing model with service time \( s(j) \) determined by:

\[
\eta'_j(k) = \frac{\lambda(j)s(j)}{1 - \lambda(j)s(j)} ,
\]

i.e. we tax the real arrival rate \( \lambda(j) \) and real mean queue length \( \eta'_j(k) \) and we compute the corresponding mean service time under the assumption of an \( M/M/1 \) queue.

- Then we may apply the formula (cf. [Kleinrock (1975)]) for the distribution function of the waiting time \( W \):

\[
P[W \leq x] = 1 - \lambda(j)s(j)e^{-(1-\lambda(j)s(j))x/s(j)}
\]

Hence the \( \alpha \)-excess point is:

\[
-\frac{s(j)}{1 - \lambda(j)s(j)} \log \frac{\alpha}{\lambda(j)s(j)} \quad (\alpha < \lambda(j)s(j)) .
\]

We continue with some remarks on extensions. If we assume that external trucks and slave trailers have different service times, we may modify the algorithm in (7) by replacing \( s(y(j)) \) by the service time of external trucks.

Another extension is to consider priority rules for slave trailers and external trucks. If we want to give slave trailers priority over external trucks we must replace \( \eta'_j(k-e_j) \) by zero in (3). If we want to give external trucks priority over slave trailers we must replace \( \eta'_j(k) \) by zero in (7). The priority cases are not exact but approximations.

We conclude with a simple example. Consider the situation (a) and (b) of figure 6 (page 16).

In case (a) the slave trailers are going from shipcrane \( S_1 \) to yard stacker \( Y_1 \) and from \( S_2 \) to \( Y_2 \).

In case (b) the slave trailers connected to \( S_1 \) go to \( Y_1 \) and \( Y_2 \) randomly and the same holds for the slave trailers connected to \( S_2 \).
In case both shipcranes have the same number of slave trailers and the service times and travel distance are equal it is intuitively clear that (a) is better than (b). The model makes it possible to compare several different organization forms. In principle it is possible to use this model for operational decision making about the assignment of slave trailers: when there is a change in the parameters \( \lambda(j) \) or in the sets \( I \) and \( J \), then the operational manager may search for the assignment of slave trailers that has the best performance, according to one of the criteria.

5. Models for the behaviour of the whole terminal

In [van Hee et al. (1986)] we treated a model for the computation of the mean and variance of the storage utilization. This model is also applicable for the stack yard and the utilization of the stack yard is an important parameter to compute the net-service times of the yard stackers (cf. section 3). We recall the formulas: the mean and variance of the number of containers on the stack yard are respectively:

\[
\mu = \lambda \cdot \mu_B \cdot \mu_G
\]

\[
\sigma^2 = \lambda \cdot \mu_B \int_0^\infty (1-G(x))G(x)dx + \lambda (\mu_B^2 + \sigma_B^2) \int_0^\infty (1-G(x))^2 dx ,
\]

where:
A is the arrival rate of ships
B is the distribution of the number of containers on the ship (only $\mu$ and $\sigma^2$ are required)
G is the distribution of the residence time of a container on the yard. ($\mu$ and $\sigma^2$ denote mean and variance)

We may apply these formulas for the import and export containers both.

In [van Hee et al. (1985)] also a model for quay or berth utilization was given.

We considered there a model for two-phase service distribution and an inhomogeneous Poisson arrival process with a periodic arrival rate (the period is typically a week). We discretized the model where the time unit was chosen such that the probability of two transitions was neglectable (typically about an hour). The results of the model are the distribution of the number of ships in each phase at the quay for each time interval. Note that the mean service times of the shipcranes may be different in each time interval due to arrival rate of external trucks in that time interval.

The two phases of ship service are: unloading of import containers and loading of export containers.

To compute the berth occupation we need the throughputs of the shipcranes and the arrival rates of the ships. The first ones vary with each hour of the day, the second ones with each hour of the week. With these numbers we have the transition matrices for each hour and so we may compute the distributions by matrix multiplication, given a start distribution (for instance empty). It turns out that the distributions over a day stabilize quickly: after a few 'weeks' the distributions over a day of the week do not differ much.

The throughputs vary for each $(k,l,m)$ where $k$ is the number of ships in the import phase, $l$ the number in the export phase and $m$ the hour of the day. To compute these throughputs with the model of section 4, we have to know the arrival rates of external trucks and the 'organization form'.

It is assumed that the arrival pattern of trucks is the same for each day. The organization form is an assignment of slave trailers to shipcranes and to yard stackers. It is assumed that for each $(k,l,m)$ such an assignment
is given by the decision maker. Of course, one may search with the model of section 4 for a 'good' assignment for each situation. The integration of models is illustrated in a dataflow diagram in section 6.

6. Architecture of the dss

First we recapitulate the scenario parameters of the system:

S1: arrival rates of ships for all hours in a week
S2: means and variances of the number of import and export containers of a ship
S3: residence time distribution of import and export containers
S4: arrival rates of external trucks for all hours on a day
S5: yard stacker parameters
S6: shipcrane parameter: mean service time
S7: yard parameters (length, width, maximal height, export and import areas, etc.)
S8: travel distances on the yard
S9: number of berths
S10: assignment rules for yard stackers to shipcranes and slave trailers to shipcranes for each three tuple \((k, l, m)\) (cf. section 5).

In figure 7 we give the dataflow diagram of the dss. The transformation processes are:

T1: storage utilization: mean and variance of the number of containers in import and export compartments (see section 5)
T2: yard stacker model (see section 3)
T3: network of queues representing the interaction of equipment (see section 4)
T4: berth utilization model (see section 5).

The loop from T3 to T2 must be initiated by an estimate of the arrival rate of slave trailers. Note that this arrival rate is of influence of the
Figure 7. Dataflow diagram.
yard stackers service time and that this service time is a parameter for
the queuing network which produces the arrival rate of slave trailers.
So here an iteration procedure is necessary. The dotted line indicates
that lost calls may be used to adapt the arrival rate of ships.
The dss has a database for the storage of all scenario parameters and
computed values.
The user interface is menu-oriented. The user may change all scenario
parameters. He may initiate every transformation process provided the
parameters are available.
The system may be used for several purposes, for instance to optimize a
yard stacker or to design an assignment rule for slave trailers. So one
does not want to start with $T_1$ always. Therefore the computed values from
the transformation process can be stored.

7. References

Baskett, F., Chandy, C.M., Muntz, R.R., Palacios, F.G.: "Open, Closed and
Mixed Networks of Queues with Different Classes of Customers",
J.ACM 22 (1975), 248-260


Reiser, M., Lavenberg, S.S., "Mean Value Analysis of Closed Multichain

Van Hee, K.M., Huitink, B., Leegwater, L., "Portplan, decision support
system for port terminals", Designing Decision Support Systems

Wolff, R.W., "Poisson arrivals see time averages", Oper. Res. 30 (1982),
223-231.

Zahorjan, J., Wong, E., "The solution of separable queuing networks using