Special Issue *SIMULATION on Harbour and Maritime Simulation*, 1998, to appear

Simulation and Planning of an Intermodal Container Terminal

Luca Maria Gambardella, Andrea E. Rizzoli, Marco Zaffalon
IDSIA - Istituto Dalle Molle di Studi sull'Intelligenza Artificiale
Corso Elvezia 36, 6900 Lugano, Switzerland
{luca,andrea,zaffalon}@idsia.ch

Abstract

A decision support system for the management of an intermodal container terminal is presented. Among the problems to be solved, there are the spatial allocation of containers on the terminal yard, the allocation of resources and the scheduling of operations in order to maximise a performance function based on some economic indicators. These problems are solved using techniques from optimisation, like job-shop scheduling, genetic algorithms or mixed-integer linear programming. At the terminal, the same problems are usually solved by the terminal manager, only using his/her experience. The manager can trust computer generated solutions only by validating them by means of a simulation model of the terminal. Thus, the simulation tool also becomes a means to introduce new approaches into traditional settings.

In the present paper we focus on the resource allocation problem. We describe our modules for the optimisation of the allocation process and for the simulation of the terminal. The former is based on integer linear programming; the latter is a discrete event simulation tool, based on the process-oriented paradigm. The simulator provides a test bed for checking the validity and the robustness of the policy computed by the optimisation module. The case study of the Contship La Spezia Container Terminal, located in the Mediterranean Sea in Italy, is examined.

1. Introduction

The management of an intermodal container terminal is a complex process that involves a vast number of decisions. Most of the world's goods which are traded daily are transported via intermodal terminals. Goods arrive and leave on various transportation means such as trucks, trains and vessels. An intermodal container terminal plays a fundamental role in routing goods to and from their origins and destinations. It is a basic node in a transportation network, where thousand of daily decisions are taken to manage this sustained flow of containers.

The advent of management information services and data processing greatly improved the ability of terminal managers to control the whole process, but still raw data has to be analysed and treated to provide some insight on the performance of terminal operations. Simulation models have proven to be a reliable and convenient tool to support the decision makers in the daily operations in many cases (Hayuth *et al.* 1994, Blümel, 1997, Bruzzone and Signorile, 1997). They provide a test-bed to assess the validity of management policies and can be used to point out problems such as conflicts in resource allocation and terminal space management. These simulation tools do not provide answers to question such as "how can I minimise the time it takes to unload these two incoming ships?" or "Should I unload the ship, or wait for the train to arrive?". In many cases, these answers are yet to be provided by the terminal managers, basing their decisions on experience in solving these problems.

A substantial help to terminal managers can derive from Decision Support Systems (DSSs) where planning and management techniques, derived from the Operations Research and Artificial Intelligence fields, can be coupled with simulation models and statistical data analysis tools. The role of simulation becomes of paramount importance in such a setting: human decision makers tend not to trust computer generated management policies, unless they either fully understand the way they were generated or are provided with sufficient evidence of their validity. This behaviour is often proven to be reasonable, since very often computer generated policies are not flexible enough in comparison to the complexity and high stochasticity of real world operations.

A well designed simulation tool can be the middle ground where decision makers compare their own experience with the DSS generated management policies and validate them. Under this point of view, it is clear that the possible strength of mathematical approaches to the optimisation of terminal processes are highlighted in a proper way to terminal managers.

2. The problem

An intermodal container terminal is a place where containers enter and leave by multiple means of transport, as trucks, trains and vessels (I/O transport means). We focus our attention on the case study of La Spezia Container Terminal (LSCT), located in the Tyrrhenian sea in Italy.

Containers arrive at LSCT by train, vessel or truck and are stored in the terminal yard. Containers then leave the terminal by the same means to reach their next destinations. The flow of containers is composed of an *import flow*, i.e. containers unloaded from ships, to be either transhipped or directed to the final destinations by trucks and trains, and an *export flow*, i.e. containers loaded on ships leaving the terminal.

In the LSCT, containers are stacked up to the fifth level on the yard by rail-mounted cranes (*yard cranes*) which unload trucks and trains. This stack height is quite unusual and is due to the lack of space on the yard. LSCT is a terminal with a high traffic on a small yard and therefore the management of space is a critical issue. *Quay cranes* unload vessels and place containers on *shuttle trucks*, which move them to storage locations in the yard. Loading a vessel is a similar process, where the shuttle receives the container from the yard cranes and moves it to the proper quay.

The amount of work processed by a container terminal depends on the quantity of containers in transit.

3. Decision support for terminal management

Storing containers on the yard, *allocating resources* in the terminal, and *scheduling vessel loading and unloading operations* (L/U operations, for brevity) are major problems in an intermodal container terminal. To solve these problems we define an architecture composed of three different but strictly connected modules (Rizzoli *et al.*, 1997) (see figure 3-1):

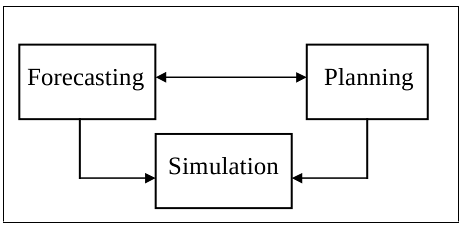

Figure 3-1 The modular system architecture

- a simulation model of the terminal, described in terms of entities (work force, transport means, storage areas, etc.) and processes (vessel load/unload, shuttle truck movements, crane operations, etc.);
- a set of forecasting models to analyse historical data and to predict future events (Box *et al.*, 1994; Vemuri and Rogers, 1993), thus providing estimates of the expected import and export flows;
- a planning system to optimise L/U operations, resource allocation, and container locations on the yard.

This architecture supports the terminal managers in the evaluation of:

- vessels loading and unloading sequences in terms of time and costs;
- resource allocations procedures;
- policies for container storage both in terms of space and cost of operations.

This allows terminal managers to assess "what-if" scenarios; for instance, what happens if the terminal undergoes an increased input/output throughput, or even if structural changes are made (e.g.: new berths are built, new cranes are added).

As the forecasting module is described in previous papers (Gambardella *et al.*, 1996, Bontempi *et al.*, 1997), in the following sections we introduce the other two modules of our architecture: the planner and the terminal simulator. For each topic, we present the major problems, the resolution methodologies and the experimental results obtained at the current state of the project.

3.1. The optimisation modules

In our study, we identified a series of problems, placed at different representation levels, which can be assisted by a computerised decision support system: spatial allocation of container locations in the terminal yard, allocation of terminal resources (yard and quay cranes, work force, etc.), and scheduling of terminal operations (e.g. container movements) in order to maximise a performance function of economical indicators. These problems also have different planning horizons related to the speed of the dynamics of the system they control: the spatial allocation policy has a horizon of about one week, while a few work shifts (about twenty-four hours) is the horizon of the resource allocation policy. The planning horizon of scheduling of terminal operations can be as short as one hour. In this paper we will focus our attention on the resource allocation module, since this is part of the case study we present later.

3.1.1. Resource allocation

The role of the resource allocation module is to solve the problem of resource allocation for vessel loading and unloading operations (Zaffalon and Gambardella, 1998). The problem of resource allocation can be formulated as a mixed integer linear programming problem (MILP) with the goal of maximising the profits over a discrete and limited time horizon (the time is discrete since the shift, i.e. 6 hours, is the minimum unit of time for resource allocation). The objective function depends on the costs of resource usage, the lateness in vessel loading/unloading and the income of the terminal for each operation. The algorithm we have implemented accepts as inputs the list of scheduled ships, their estimated time of arrival, the forecasted number of containers to be loaded and unloaded and the yard regions involved in loading and unloading operations. The outcome is a schedule of the yard and quay cranes employed in the upcoming work shifts together with a deterministic forecast of expected profits. The simulation tool is employed in order to validate the solution provided by the algorithm. The simulator is also used for comparing such a solution to the allocation implemented at LSCT by the decision makers.

The resource allocation module is designed to model the terminal as a flow network where resources are used to distribute containers from ships to different yard areas and viceversa. This module is designed to determine the amount of resources needed to perform a set of terminal operations. It is important to note that the resource allocation module uses the same information available to the terminal managers performing the same task by hand. In fact, before a ship’s arrival, the available data are: the expected time of arrival of the ship, the total number of containers to load and unload, the container distribution on the terminal yard and the allocation of import and export areas in the yard.

Below we give a brief description of the MILP model.

As far as resource allocation is concerned, the terminal can be interpreted as a mechanism allowing the container flows to be directed from their sources to the proper destinations, with the objective of maximising the profit. This view takes the terminal model close to a flow network (Papadimitriou and Steiglitz, 1982). Let us see how the terminal can be mapped to a flow network with a simple example. Consider a single shift and suppose that the only ship at the terminal is ship S, parked at quay 1. Suppose that ship S must only be loaded, with 100 containers that are already in the yard according to the following distribution.

Table 3-1 Containers Distribution for S

Ship Name	Quay	CA	CB	CC	A	R
S	1	10	15	30	0	45

Table 3-1 shows how the amount of containers related to S is distributed in the yard areas (CA, CB, CC, A and R). In particular, 10 containers are stored in CA, waiting to be moved into the ship; other 15 containers are waiting in CB; 30 containers are in CC and 45 in R. Such containers must go from the areas to the ship passing through the quay cranes of bay 1. We describe this process by means of the graph in figure 3-2.

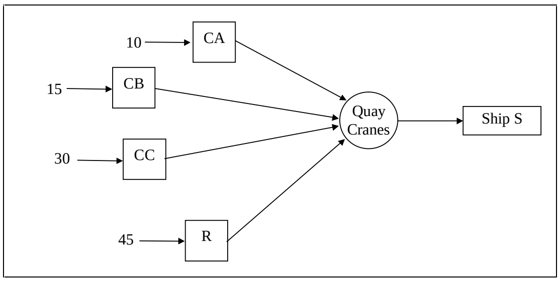

Figure 3-2 Flow network for ship-S example

The diagram in figure 3-2 is a graph where the square nodes are yard areas (the node corresponding to A is not present, since there are not containers in area A for S), the circle stands for the quay cranes serving ship S, and the rectangular node is for the ship itself. An arc from node X to node Y, means that part of the total flow in the network can run from X to Y. For instance, the path CA -> Quay Cranes -> Ship S, denotes that the containers stored in area CA have to be moved to the quay cranes and (only) then to ship S. In the above network, the square nodes are sources of flow. This is highlighted by their incoming arrows, that generate a certain amount of flow coming from outside the net. Instead, the rectangular node is a sink of flow, since no arc departs from it.

The flow network in figure 3-2 is a representation of the possible paths of containers moved inside the terminal. The representation is completed by the introduction of arc capacity. Arc capacity is the maximum number of containers that can run through an arc (in the shift). In figure 3-3, we place a label near every arc to denote its capacity.

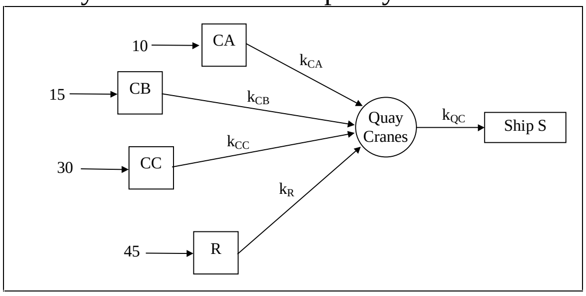

Figure 3-3 Arc capacities

Thus, for example, k_{CA} stands for the maximum number of containers that can be moved from CA to the quay cranes in the actual shift. In the same way, k_{QC} is the maximum number of containers that can be moved by the quay cranes to the ship in the actual shift. It is clear that arc capacity is a function of the resources that are allocated in the shift. Since the goal of the model is the computation of the allocation plan, it follows that arc capacities are not constants, depending on the number of resources that the decision maker allocates. In other words, we are dealing with a flow problem where the main decision coincides with the allocation of resources, which then determines the capacities of the arcs in the flow network. This makes the model a particular case of a network design problem (Magnanti and Wong, 1984).

The objective of the described model based on the flow network is the maximisation of the profits, where the latter is the difference between the income and the expenses. The income is a term proportional to the number containers moved into the ship. The expenses, in the LSCT case, are a linear function of the allocated resources.

A major observation regarding the flow model of the terminal as described above is that the optimisation problem is linear. In fact, flow constraints are linear, and the objective of the terminal formulation is linear for definition. This constitutes a bridge between the formalisation of the terminal and the known resolution methods for linear programming, like the simplex method and the branch & bound. The latter is used for (mixed) integer linear programming formulations and is needed for the present flow problem, because some quantities can only be defined as integer variables (this is the case of the allocation variables, since they represent physical units).

The cited example shows that a simple case of resource allocation in the terminal is suitable to be modelled by a network design problem. Our complete model is based on the flow view of the terminal, where the flow model is greatly extended in order to take into account a number of issues: the modelling for a period of arbitrary length; a general number of scheduled ships in the terminal, including ships to be worked in parallel and queued ships; ships that must be both loaded and unloaded; ships with fixed or variable deadlines for the service; ships with limits on the number of quay cranes that can work on them; an homogeneous treatment of trains and trucks.

The resulting model is a complex mixed integer linear program, whose solution is demanded to a MILP solver. In particular, our software module for resource allocation exploits the branch & bound capability of LP_SOLVE (Berkelaar, 1998) to carry out the computation of a *good* feasible solution. In fact, getting optimality is a time-consuming task in our experience, since the complexity of the problem produces a huge search tree. Notwithstanding, a good solution is generally found quickly (few minutes on a pentium 133 Mhz, 32 Mbytes of ram); furthermore, experimental results based on real cases of the LSCT show that the best value found is usually *close* to the continuous bound.

3.2. The simulation module

The architecture of the simulation tool is based on the partition of simulation objects between simulation agents and simulation components. In an intermodal terminal there are two parallel flows: information and containers; the simulation agents use the flow of information to make decisions on how to direct the container flow.

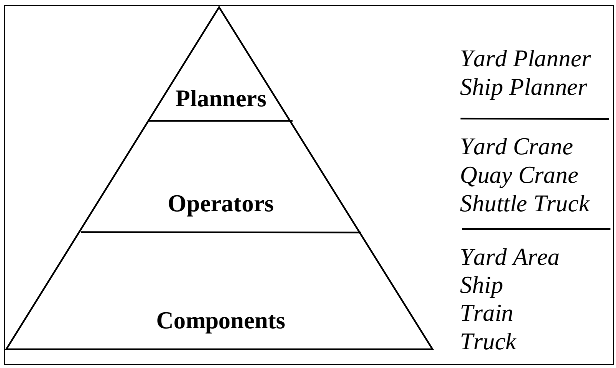

Figure 3-4 The hierarchy of simulation agents

We founded the design of the simulation tool on the object-oriented analysis and design paradigm (Booch, 1994), we modelled simulation agents and components as objects which store and exchange

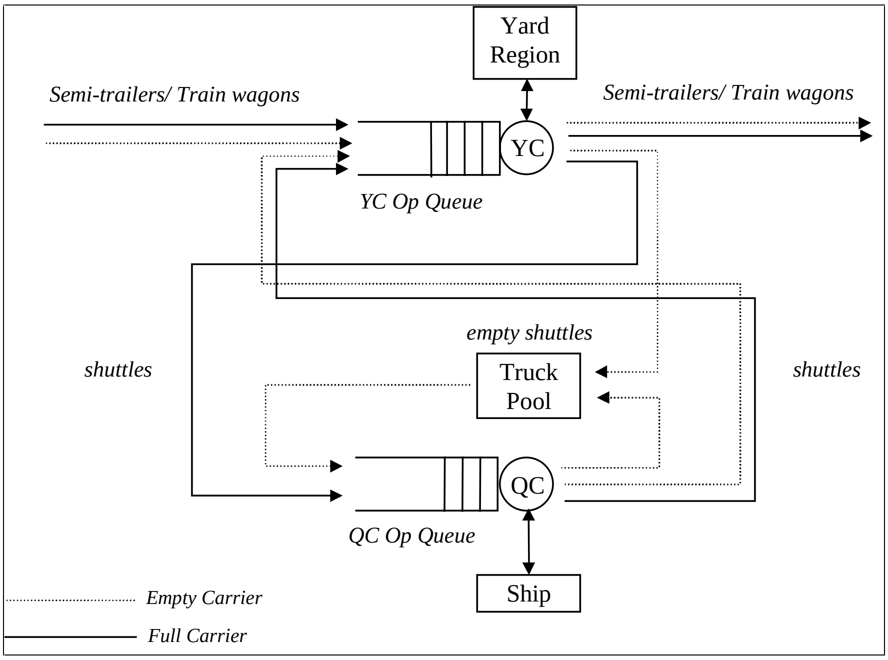

Figure 3-5 A schematic representation of the flow of containers

information on terminal inputs, states and outputs and which perform actions according to their local behaviour. There is no unique supervising agent which controls the whole simulation, but the simulation is the result of the interaction of single agents, each one endowed with "local" knowledge on its actions in response to the behaviour of other agents (Zeigler,1984 and 1990).

There is a hierarchy of simulation objects according to their "intelligence" (see figure 3-4). Planners, such as yard and ship planners are at the top, since they take the informed decisions on resource and space allocations we were concerned of in the optimisation section. Crane operators (yard and quay) and shuttle truck drivers, occupy the middle layer since they have the local knowledge which allows them to perform container movements, avoiding local conflicts, such as two yard cranes competing to place containers in the same yard area. At the bottom, there are the terminal components, such as yard areas and the containers and other agents such as ships, trains and trucks, which in principle are "intelligent" but that were modelled as "dumb" since their behaviour is imposed as an external constraint and not directly controllable by the terminal operator.

The simulation tool replicates the terminal activities and it is based on the principle that external events generate responses by the simulation agents which in turn operate on simulation components. The responses of simulation agents are determined according to the policies which can either be generated by the optimisation modules or by a representation of the experience of terminal operators.

External events are: trucks arriving at terminal gate; ships arriving at terminal pier; trains arriving at terminal. The arrival generator is a part of the simulation tool which generates these arrivals. Ship and train arrivals are read from a database, since they are known in advance, while truck arrivals are generated according to statistical distributions.

In figure 3-5 we report an example of the container flow in the terminal limited to a quay crane (QC) and a yard crane (YC). In the real terminal there are seven quay cranes and twenty yard cranes. Ships, trains and trucks entering the terminal have a loading list, the containers to import, and an unloading list, the containers to export (the lists are composed by only one element in the truck's case). These lists are used by the yard and ship planners.

The ship planner is a simulation agent dedicated to organise the loading and unloading operations of a ship. The ship planner performs the following tasks:

1. Allocate the quay cranes work shifts needed to load and unload the ship, given the ship import and export list. This task can be performed either using the resource allocation optimisation module or by entering the resource allocation strategy decided by the human operator.
2. Compute the bay plan. In general, unloading occurs before loading, and these two activities must respect the ship structural stability constraints (Sha, 1985), these constraints result in

the work order of the bays. At present the bay plan is simply computed assuming that the quay cranes progress in a constant direction while operating on a ship (e.g. from left to right). In a more sophisticated and realistic version of the ship planner, the bay plan should be computed by the scheduling algorithm which is in charge of preparing the work lists of the quay cranes.

3. Ask the yard planner to assign destinations in the yard according to the containers to be unloaded. These containers are unloaded in order as stowed. The unloaded containers will be stored in sub-regions of the yard areas, named import areas. The size and location of import areas is a decision variable.
4. Communicates to the yard planner the containers to be loaded. This list is ordered by a set of constraints which imposes a sequence to be respected in stowing containers aboard according to their size, weight, port of destination, and to a series of distinctive characteristics such as hazard class, kind of good transported, etc.
5. Put the quay cranes to work according to the plan previously determined. Supervise loading and unloading operations, collect statistics and evaluate performance.

The lists of import and export containers (see items 3 and 4 in the previous numbered list) are used by the yard planner simulation agent to build the schedule solving the job-shop problem associated with yard crane operations. The development of such an algorithm is the focus of the current and future research and it will be a critical part of the system since it will produce either the bay plans and the worklists for the quay cranes besides the scheduling of yard operations. In the current version of our system we have devised a simple algorithm which "buffers" containers which cannot be moved for lack of resources rescheduling them for a later time, when the needed resources (yard and straddle cranes) will be made available.

Another major task of the yard planner simulation agent is to organise the container allocation on the yard in order to maximise yard crane performance, avoid crane deadlocks (which could happen when two cranes try to work on the same yard area), and minimise the time to access containers during storage and retrieval. In detail, its tasks are as follows:

1. Allocate the yard cranes work shifts, given the list of containers to be loaded and unloaded by all the ships and trains that are present or are due to arrive.
2. Organise the yard space according to a given policy, selected among one of those assessed with the optimisation module (automatic parking).
3. Solve the job-shop scheduling problem, using the available data on trains, trucks and ships to be loaded and unloaded. The result is the work list (the ordered list of containers to be moved) for each yard crane and for each quay crane. These work lists are computed using a scheduling algorithm implemented in the optimisation module. Because of the stochasticity of the terminal processes, the task of the scheduling algorithm is to produce a feasible solution given the initial conditions. During terminal operations, at recurrent intervals, it will be necessary to run reactive scheduling algorithms to face unexpected changes such as delays and crane downtimes.
   In the present implementation, this centralised scheduling policy is replaced by a distributed policy generated by local rules used by crane agents. Besides this high level management performed by the ship and the yard planners, there are the local management decisions taken by "less intelligent" simulation agents such as cranes and shuttle trucks.

Quay cranes start to work when the ship planner allocates them in a work shift and assigns them a list of containers to be loaded and unloaded (the worklist). Quay cranes stop working when they have finished to process their lists; even if their work shift is over they continue to process their work list, thus accruing overtime. Quay cranes move containers to and from shuttle trucks which run between the quays and the yard cranes. When the quay crane unloads a container, it asks the yard planner which yard crane is assigned to it, the truck will therefore travel to the yard area where that yard crane is working. While the yard locations of containers are pre-determined, the yard crane which will move them is not assigned until the container is to be moved. This is the current choice of implementation, given the lack of a scheduling algorithm which would create worklists for the yard cranes.

Yard cranes pick up and put down containers on the yard. They have a queue of operations to be performed. An operation is a container movement, either picking it up from a truck and placing it on the yard or vice versa, and even temporary moves to unpack stacked containers are operations. As we have seen before, this queue of jobs (the work list) can be automatically optimised by a job-shop scheduling, or can be managed by local rules, which try to emulate the behaviour of the human operator. Currently, the “operation to server” (container to yard crane) assignment policy is to place the operation (fetch a container from the yard or storing it) on the yard crane where the distance to the storage position of the last container in the operation queue is minimal. This ensures that yard cranes avoid travelling long distances on the areas without moving containers.

Yard cranes are also provided with tie-breaking mechanisms to avoid deadlock: it can happen that, given the randomness associated with the time a crane moves a containers, the job queues push the cranes towards a conflict, such as trying to move two containers which are stored in the same bay in the same time. The yard cranes can acknowledge this potential deadlock and reassign one of the container moves to contiguous crane (this is a sub optimal solution, but avoids computing again the whole job-shop problem).

In figure 3-6 we report a typical screen-shot of the terminal during a simulation. A ship is moored on the west pier (north is to the left of the picture) and it is being unloaded by two quay cranes QC1 and QC2 (only QC1 is active, though). Containers are to be positioned on yard areas CA, CB, and CC. On these yard areas the yard cranes YC from 1 to 9 are working.

In figure 3-7 a series of histograms which are updated on-line is reported. Each histogram is associated with a yard crane and it reports the number of containers which have waited from the moment they joined the queue of truck waiting to be unloaded to the moment they were placed on the yard.

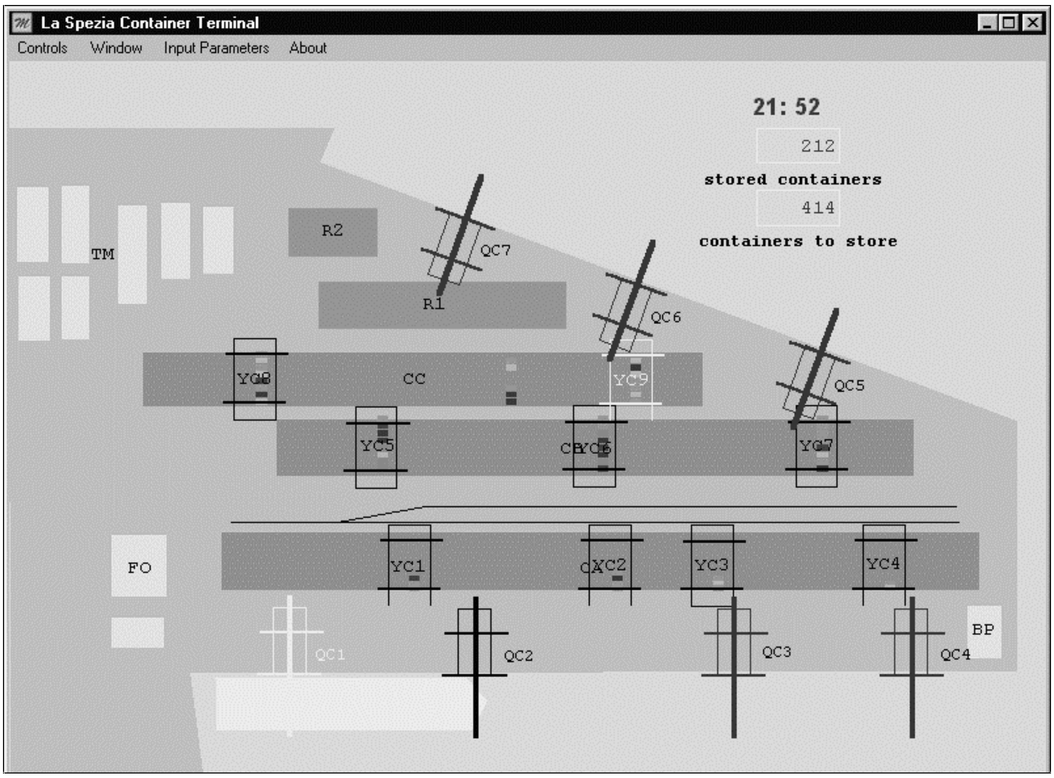

Figure 3-6 The simulation tool user interface

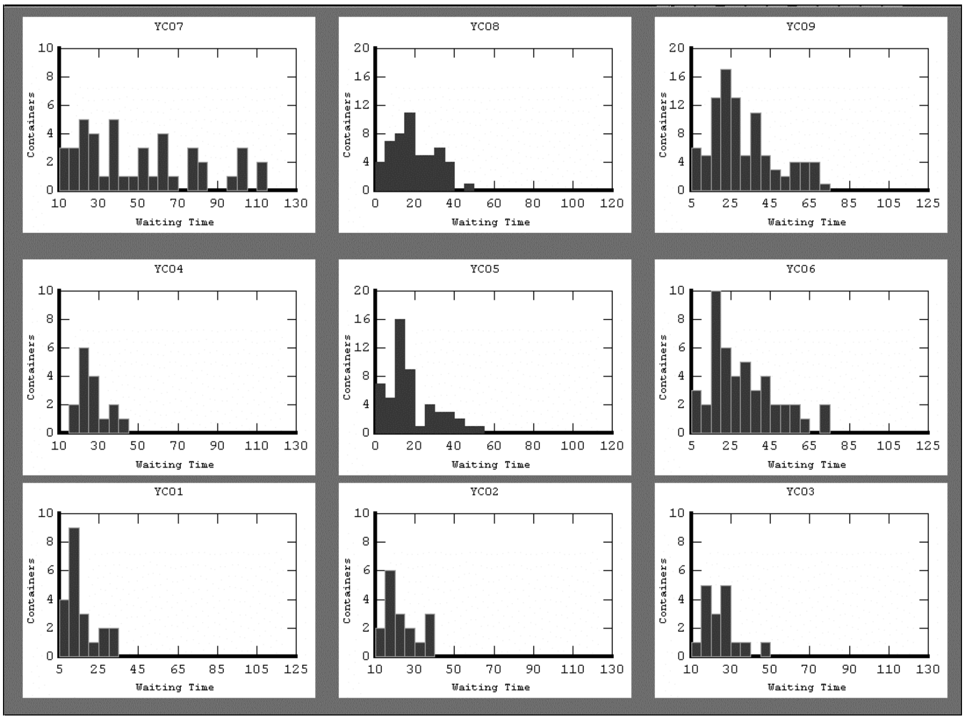

Figure 3-7 Histograms representing waiting times in front of yard cranes for containers to be served

After the completion of the simulation tool we have validated it against a set of real data and scenarios. Firstly, we have compared its behaviour with the one of the real terminal. In particular, we have compared the amount of time needed by the terminal to load/unload a given set of ships with the time required by the simulator to perform the same set of operations. Results show that our model is close to the real terminal behaviour in term of the average length of the operation queues on quay cranes.

4. Simulation for decision support: the case of resource allocation

The simulator is used as a test bed to assist the management in comparing the computer-generated resource allocation policies with their own experience. We will show how the computer-generated policies can be used as a decision support tool to produce actual resource allocation decisions.

Starting from the same terminal state (i.e. the container positions on the yard) and given the same expected inputs (i.e. the ship and train arrivals and their expected loading and unloading lists, the truck rate of arrivals), the performance of various resource allocation policies are compared, using the same job scheduling policy. We are interested in evaluating the performance of the various policies, both computer and human generated, against our working model of the terminal. The indicator used to compare the solutions is the net profit of the terminal operations during the simulation horizon. This indicator takes into account the cost of cranes and operators during the various work shifts, the penalty to be paid to the shipping company if the ship departure is delayed and the income generated by each container loaded and unloaded from a ship.

Simulations are repeated in order to collect a number of instances of performance indicators in order to experimentally measure the robustness of each policy in the face of stochastic perturbations of parameters such as the time taken by shuttle trucks to move from crane to crane, the time employed by yard and quay cranes in bay to bay movements and in container handling operations and so on (an introduction to output analysis can be found in Banks et al., 1996).

The present section describes the analyses conducted on a real case, related to the operations on three ships that arrived in the time span of about 24 hours at the LSCT. We focus on the application of the computer-generated allocation plan to the simulator, in order to evaluate the allocation policy and examine its robustness.

The data that we use are extracted from the database of the LSCT. Such data provide a description of the

Table 4-1 Scheduled ships at the LSCT

Name	Quay	Eta	Deadlines	CA	CB	CC	A	R
Morelos	2	10/11/96 16.50	2	366	93	76	67	400
Rhein Trader	3	10/11/96 23.00	2	14	46	1	0	0
Nll Korrigan	4	11/11/96 01.10	5	23	131	331	3	940

Table 4-2 Output of the allocation module

Date and Time	Shift	Pier West Cranes	Pier East Cranes	CA Cranes	CB Cranes	CC Cranes	A Cranes	R Carts
10/11/96 13.00	0	1	0	2	0	0	0	1
10/11/96 19.00	1	4	1	4	2	2	0	6
11/11/96 01.00	2	3	3	4	2	2	2	6
11/11/96 07.00	3	0	3	1	1	2	0	9
11/11/96 13.00	4	0	3	0	1	2	0	8
11/11/96 19.00	5	0	2	1	2	2	0	2

LSCT at the level of the single container: it is possible to trace the path followed by a single container and also every action of the resources. The queries to the database allow the schedule in table 4-1 to be extracted.

The above data describe the scheduled ships during the considered period, a row corresponding to a ship. In particular, the first column is for ship's name. Column "Quay" denotes the quay where a ship is expected to arrive to the LSCT. "Eta" means expected time of arrival and gives the date and time for the scheduled ship. The following column is for the treatment of deadlines. A deadline is the last shift that can be used for the service of the corresponding ship (shifts are numbered from zero on). Finally, the last 5 columns describe the distribution on the yard of the containers for a ship, in the same way as table 3-1 does for the example ship S.

The data in table 4-1 are the basis on which the resource allocation module produces the allocation plan that is shown in table 4-2.

Table 4-2 lists the resources to be allocated in the different work shifts. In particular, a row corresponds to a shift (the start time of the shift is given in the first column); for example, the first row (shift 0) describes the following allocation: 1 quay crane at pier west, 2 yard cranes in CA and 1 cart in R. According to the resource allocation module, the above plan allows all the ships to be completely served, obtaining the net profit of about 190 units. The quality of the solution is good as compared to the continuous bound provided by the solver, that is 197 units.

This resource allocation plan is then fed into the simulator. After running a few simulations some results emerge: the plan is acceptable with respect to the ship deadlines, they are always respected for the first two ships, while the third ship sometimes requires the allocation of an extra straddle crane (cart) in the last shift (n.5) to be robust to stochastic parameter perturbation. Moreover, the simulation reveals that during shift n. 4 and 5 the yard cranes on area CC have moved a very low number of containers. This suggests that their allocation can be subject to revision. Performing a novel set of simulations reducing the number of cranes on CC in those shifts shows some hint that the hypothesis was reasonable. Currently, the results of the resource allocation module are taken as input by experienced operators that use the simulator to validate and improve the proposed allocation scheme.

These improvements are sometimes possible due to the fact that our ship loading list generator implemented in the simulator is not very sophisticated and it is not able to optimise the use of the allocated resources. We are now working to improve this module by introducing a new loading list generator based on flexible job shop scheduling (Mastrolilli and Gambardella, 1998).

5. Conclusions

In this paper we have shown the use of optimisation and simulation as decision support tools in the management of a real world intermodal terminal. We focused our attention on the problem of resource allocation and have shown how operation research techniques can be used to generate resource allocation plans which can be used to support the terminal managers in deciding their management strategies. For this purpose, the simulation tool, based on a more realistic representation of the terminal than the one used in the model adopted to solve the resource allocation problem, is a fundamental tool to assess the validity of the computer generated solution, to modify it and to put it in relation with the real world.

Our current work is aimed at providing another decision support tool which we deem fundamental to improve terminal management: a job-shop algorithm which could generate the import and export stowage plans for each ship and train entering and leaving the terminal which would have to be coupled with a shorter-term reactive job-shop module which could manage the work sequences on each crane in the terminal.

Acknowledgements

This project is supported by the Swiss Commission for Technology and Innovation under the project 3128.1 “A Methodology for Containers Flow Forecasting and Positioning in Intermodal Terminal” a join effort between IDSIA and Data and System Planning SA, Manno, CH.

References

J. Banks, J.S. Carson, B.L. Nelson, *Discrete-Event System Simulation* - 2nd edition, Prentice Hall, Upper Saddle River, New Jersey, 1996.

M. Berkelaar, LP_SOLVE, ftp://ftp.es.ele.tue.nl/pub/lp_solve, 1998

E. Blümel, (ed.), *Managing and Controlling Growing Harbour Terminals*, Society for Computer Simulation International, 1997.

G. Bontempi, L.M. Gambardella, A.E. Rizzoli, Simulation and optimisation for management of intermodal terminals, paper presented at ESM ’97, Istanbul, June 1-4, 1997.

G. Booch, *Object-oriented Analysis and Design: with Applications, 2nd Edition*, Rational, Santa Clara, 1994.

G.E.P. Box, G.M. Jenkins, G.C. Reinsel, *Time Series Analysis: Forecasting and Control*, Prentice Hall, London, 1994.

A.G. Bruzzone, R. Signorile, Genetic algorithms and simulation as support for planning a port terminal, paper presented at Simulation in Industry, 8th European Simulation Symposium, Genoa, Italy, 1996.

CACI Products Company, *MODSIM III, the Language for Object-Oriented Programming, User’s Manual*, La Jolla, 1996.

L.M. Gambardella, G. Bontempi, E. Taillard, D. Romanengo, G. Raso, P. Piermari, Simulation and forecasting in intermodal container terminal, paper presented at Simulation in Industry, 8th European Simulation Symposium, Genoa, Italy, 1996.

Y. Hayuth, M.A. Pollatschek, Y. Roll, Building a port Simulator. *Simulation*, 63(3), 179-189, 1994.

T.L. Magnanti, R.T. Wong, Network design and transportation planning: models and algorithms. *Transportation Science*, 18(1), (1984), 1-56.

M. Mastrolilli, L.M. Gambardella, New Neighbourhood Function for the Flexible Job Shop Problem, *IDSIA Technical Report n.* 5-98, 1998.

H. Papadimitriou, K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, Prentice-Hall, N.Y., 1982.

A.E. Rizzoli, L.M Gambardella, G. Bontempi, Simulation of an Intermodal Container Terminal to Assist the Management in the Decision Making Process, in *Proceedings of MODSIM 9, International Congress on Modeling and Simulation*, Hobart, Tasmania, 1997.

O.P. Sha, Computer aided on board management, in P. Banda and C.Kuo (eds.) Computer Applications in the Automation of Shipyard Operation and Ship Design V, Elesevier Science Publishers, 1985.

V.R. Vemuri, R.D. Rogers, *Artificial Neural Networks: Forecasting Time Series*, IEEE Computer Society Press, 1993.

B.P. Zeigler, *Mutifacetted Modelling and Discrete Event Simulation*, Academic Press, 1984.

B.P. Zeigler, *Object-oriented Simulation with Hierarchial Modular Models: Intelligent Agents and Endomorphic Systems*, Academic Press, 1990.

M. Zaffalon, L.M. Gambardella, A network design approach to the allocation of resources in an intermodal terminal, *IDSIA Technical Report n. 10-98*, 1998.