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

Managing resource allocation, scheduling and simulation for an intermodal container terminal

L. M. Gambardella, M. Mastrolilli, A. E. Rizzoli, M. Zaffalon

IDSIA—Istituto Dalle Molle di Studi sull'Intelligenza Artificiale

Galleria 2, CH-6928 Manno, Switzerland

{luca,monaldo,andrea,zaffalon}@idsia.ch

Abstract

An intermodal container terminal is a complex system that manages the process of container loading and unloading exchanging transport modes. In particular, an intermodal sea-port moves containers from the yard to the ships and vice versa. A set of resources, such as yard and quay cranes, and straddle carriers, direct the flows of containers. Minimizing the resources needed to serve the workload is the most critical task in a port, because resources determine the terminal costs. It is also a difficult task, partly because it must be addressed for a relatively long time horizon. Here, the resource allocation problem is formulated as the problem of optimally designing a network of flow. Approximate solutions are obtained by formalizing the problem as a mixed-integer linear program, and by partially searching the solutions space by the method of branch and bound. This method was applied to a set of real world instances from the La Spezia Container Terminal, in Italy. On average, the solution was 6.6% far from the continuous relaxation bound; moreover, the cost of the computed allocation plans was 67% of the costs originally planned by the terminal managers. The actual sustainability of the computed plans was verified by designing and implementing both a scheduling algorithm, that optimizes the ships loading/unloading list, and a detailed discrete-event simulator of the terminal. The test system executes a loop: the computed allocations are passed to the scheduler that produces the list of container moves; on this basis, the simulation model updates the terminal state that is then returned to the resource allocation module. The system was able to serve the real workload for one week, respecting all the ships' deadlines, thus validating the proposed approach.

Key words. Intermodal terminal; Resource allocation; Optimization; Scheduling; Simulation; Network design; Network flow.

1 Introduction

Intercontinental cargo transport has been continuously increasing since the advent of container shipping in the 50s. Nowadays 95% of world cargo is moved by ship. Ports play the role of exchange hubs where containers are moved from ships to trains and trucks and their efficiency is fundamental to sustain the mounting cargo traffic. To cope with increased traffic demands and with decreasing profit margins the terminal operators need to improve the management of terminal processes such as ship berthing, ship loading and unloading, straddle crane routing, resource allocation, yard space management.

All of these problems are strictly interrelated (Section 2); for instance, the choice of the position of the containers on the yard has an impact on the decisions made regarding the allocation of resources, the allocation of resources constrains the scheduling of loading and unloading operations, and so on. The complexity of these processes makes it necessary to use efficient methods for the optimization of the overall system: operations research techniques and simulation have been proven particularly apt to solve these kinds of problems (see, for instance, Kozan and Preston, 1999; Kim and Bae, 1999; Kim and Kim, 1999).

The approach currently adopted by most terminal managers splits the problem by treating each ship as an independent entity: a human ship planner is dedicated to plan loading and unloading operations for a single ship and to allocate the needed resources in term of quay cranes, yard cranes, lifters and man power. Since there is no cooperation among different ship planners, this method is a source of conflicts and performance degradation mainly in the case of resources that must be shared among parallel loading and unloading activities. The complexity of the process leads to the separation of management activities, but even from the mathematical point of view formulating a single problem is not viable, given the complexity of the resulting model.

We propose a methodological approach (Section 3) to solve the problem which still considers the interactions among the ship planners, but reduces the problem complexity by iteratively modeling it at different levels of detail. At the highest level, container moves during loading and unloading operations are seen as a flow inside the terminal (Zaffalon and Gambardella, 1998). The goal is to determine the best *resources allocation* (RA) on the yard with the objective of minimizing the costs of the terminal. At the lowest level, given the exact position of containers in the yard, the goal is to *schedule the sequence of loading/unloading operations* in order to optimize the use of the allocated resources and to make sure that the predicted flow is sustainable (Gambardella et al., 2001). Finally, a *simulation module* is used as a test bench to evaluate the overall policies produced by the optimizations modules.

In this paper the resource allocation module is described in detail (Section 4). The problem is addressed by regarding it as a discrete-time network design (Magnanti and Wong, 1984; Ahuja et al., 1993): the port is modeled by a flow network. The capacities of the network arcs, which are a function of the allocated resources, must be dimensioned to enable the scheduled containers flows to be moved from origins to destinations while minimizing costs (Section 4.1). Approximate solutions are obtained by formulating the problem as a mixed-integer linear program, as in Section 4.2, and by searching the solutions space by the method of branch and bound (Section 5).

We have tested the overall system (resource allocation, scheduling and simulation) on two weeks of real data provided by the La Spezia Container Terminal (LSCT), in Italy. The RA module computed remarkably good weekly allocation plans in less than one minute on a 500-Mhz pentium. On average, the computed solution was about 6.6% far from the continuous relaxation bound; moreover, the computed plans allowed the amount of work of the terminal to be carried out by only spending about 67% of the resource costs originally planned at the terminal, while still respecting the deadlines imposed on the load/unload operations. It is important to emphasize that this performance was actually realized by the port simulator, when fed by the optimized policies, so highlighting the robustness of the proposed solutions in a complex stochastic environment.

2 The Intermodal Container Terminal

An intermodal container terminal is a place where containers enter and leave by multiple means of transport, as trucks, trains and vessels. In the following we consider the case of the LSCT, which is exemplified in Figure 1.

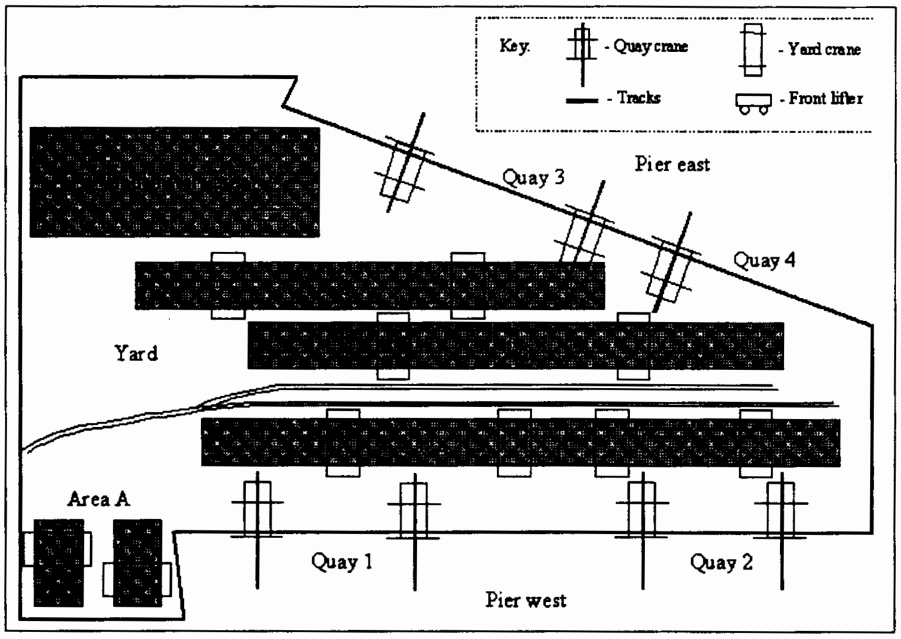

Figure 1: A schematic view of the La Spezia container terminal

The terminal is composed by the *yard*, where containers are stocked, and by the *piers*, where ships dock. Ships are served by *quay cranes*. Quay cranes on the same pier can slide on rails to serve different sections of a ship, the *bays*. The position of a container (on ship or on yard) is identified by the triple *bay*, *row* (the second planar coordinate, besides the bay), and *tier* (the position in the container stack).

Shuttle trucks move containers from the quays to the yard areas and back. The terminal yard is divided into 5 areas: CA, CB, CC, A and R, which are used for temporary storage of containers. Containers in the R area are accessed by *front lifters*. The remaining areas are served by *yard cranes*. Yard cranes can only move over the related area. Their task is to load containers into the area from shuttle trucks (the cranes in CA and CB do it also from trains) and vice versa. In LSCT there are ten front lifters and ten yard cranes (four on CA, two on CB and CC, two on A).

When a ship arrives in the terminal, it is first unloaded by a set of quay cranes. The unloaded containers are stored in sub-regions of the yard areas, named import areas, where they can be stored by shuttle trucks, by yard cranes or by front lifters. When the unloading operations are completed, the loading stage can start: containers which must be loaded on the ship are first discharged from the yard, loaded on shuttle trucks using yard cranes or front lifters, then each shuttle truck carries one container at a time to the ship where it will be loaded by using a quay crane.

3 The Methodology

The problem of *resource allocation* is the most important since the terminal costs are a function of the resources (i.e. the quay cranes, the yard cranes, the front lifters and the related operators) used to move the containers. The objective is to minimize the costs by properly allocating resources, while respecting the ships' deadlines. The solution to this problem must be found over a time horizon of one week without knowing in advance the exact positions of the containers in the yard. The detailed model and the solution procedure is reported in Section 4.

The *scheduling of ship loading and unloading* deals with the same problem but at a different level of detail. A few hours before the arrival of an incoming ship, the terminal receives detailed information about its stowage, i.e., containers that are to be stored in the yard (import containers), and about the list of containers which are to be loaded on the ship (export containers). This information allows the *ship planners* to generate the *unloading list* and the *loading list*. The unloading list specifies the order by which containers are to be unloaded from the ship in order to guarantee the ship stability (Sha, 1985). There is an unloading list for each quay crane serving the ship. Thus, the goal is to move the containers to and from the ship, but at this stage the resources have been already allocated and the container positions are finally known. Containers are usually stacked up in piles, so that computing the right sequence of operations is crucial to increase performance and to avoid deadlocks among resources.

This optimization problem has been modeled as an extension of the flexible job shop problem (Mastrolilli and Gambardella, 2000) where groups of containers are jobs, single containers are operations and resources are machines. The objective is to execute the loading and unloading operations with the given resources so that the makespan is minimized. The scheduling phase does not take into account economic factors, since the expenses have already been taken into account in the resource allocation stage. On the other hand, solving the scheduling problem with the allocated resources guarantees that the produced container flows are sustainable in practice, when we consider single container moves and not flows. The problem of scheduling loading and unloading operations is heuristically solved by local search techniques (Aarts and Lenstra, 1995) and by the tabu search paradigm (Glover, 1989). A deep investigation of our extended neighborhood function for the flexible job shop and our new self-tuning tabu search is presented in (Gambardella et al., 2001). Using the neighborhood function and tabu search in succession, it is possible to quickly (few minutes on a Pentium 266Mhz) compute scheduling policies that significantly increase the terminal performance.

Coupling *simulation* with *optimization* techniques plays a fundamental role both in testing the impact of proposed policies and to convince the terminal operators of their validity (Gambardella et al., 1998). In fact, the policies are produced under deterministic assumptions, while a terminal is clearly a stochastic environment. A simulator can act as a test bench to show that the policies are sustainable and robust. Different authors have developed discrete-event simulation models of port terminals (Hayuth et al., 1994; Young and Seok, 1999). We also have developed an object-oriented, process-oriented, discrete-event simulator of the container terminal of La Spezia in Italy to which our study is applied. The simulation model has been designed according to the object-oriented analysis and design paradigm (Booch, 1994). Simulation agents and components are modelled as objects that store and exchange information on terminal inputs, states and outputs and which perform actions according to their local behavior (Zeigler, 1984 and 1990).

There is a hierarchy of simulation objects according to the importance of the decisions they take. Planners, such as yard and ship planners are at the top, since they take decisions on resource and space allocations. Crane operators (yard and quay) and shuttle truck drivers, occupy the middle layer, and at the bottom, there are the terminal components, such as yard areas, containers, and other agents such as ships, trains and trucks, whose behavior is imposed as an external constraint and not directly controllable by the terminal operator.

The simulation module has been implemented in Modsim III (CACI, 1996), a specialized object-oriented simulation language, which supports the process-oriented view of discrete-event simulation (Erard and Déguénan, 1996). A detailed description of the simulation model we have implemented can be found in (Gambardella et al., 1998).

The *simulation model* is therefore the place where the two optimization modules are used together (Figure 2) to provide a solution to the integrated management of the terminal. The resource allocation module provides resources over horizon of one week ($RA([D_i, D_{i+6}])$, where Di is day number i), but it is re-run at the end of every day, with the updated terminal state (Current State) produced by the simulation of one day of work, that is, of four work shifts (I, II, III, IV). The allocated resources (Resources of Day i) are then

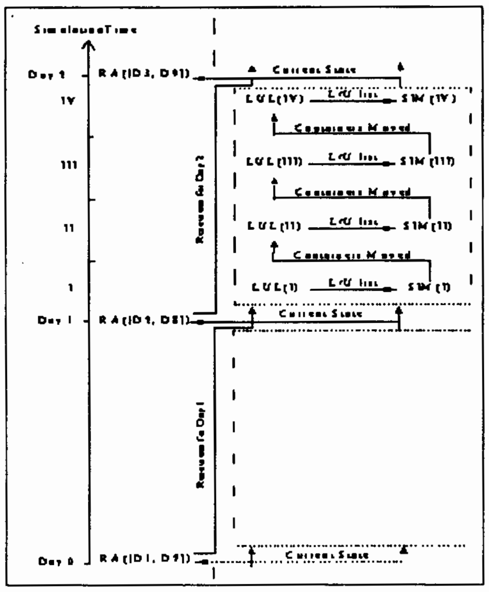

Figure 2: Schema of the simulation-optimization loop; the modules are executed in a hierarchical way, from resource allocation to scheduling and simulation

used as input by the scheduling module (LUL) that computes the loading and unloading lists (L/U list) for each crane for the next work shift. It is the simulator (SIM) that uses these lists to perform terminal operations (Containers Moved) and to update the terminal state. By repeating this "allocate - schedule - simulate" loop it is possible to assess the performance of the optimized policies in the integrated management of the terminal.

4 The problem of allocating resources

Resource allocation is made on the basis of a workplan, like that presented in Table 1. Each row of Table 1 refers to a ship.

Table 1: An example workplan that represents the arrival of two ships.

Name	Quay	Eta	Deadline	CA	CB	CC	A	R
S1	1	Shift 1 at 1:0	3	10	15	30	0	45
S2	2	Shift 2 at 7:0	3	20	5	14	10	9

The "Eta" column reports the time at which the ship is expected. This is expressed as a shift number and the hour and minutes in such a shift. A day is conventionally divided into 4 shifts of 6 hours. The "Deadline" column reports the shift by which a ship must have been served. The last 5 columns display the yard distribution of all the containers to be moved for a ship. We do not distinguish between the inflow and the outflow because the allocation of resources is not concerned with the direction of moves.

A standard workplan describes a temporal horizon of one week (28 shifts). The workplan is a forecast that involves a degree of uncertainty that increases with time. For this reason, the resources are allocated only for the following day and the one-week time window is updated daily. The time window is needed to balance the short-term and the mid-term information.

An algorithm solves the resource allocation problem by providing a set of resources for each shift in the workplan, as in Table 2. Now each row represents a shift: e.g., the first row suggests to allocate 1 quay crane at Pier West, 1 yard crane in CA, 1 in CB, 1 in CC and 2 front lifters (area R).

Table 2: An example of allocation of resources for the workplan in Table 1

Shift	Pier West	Pier East	CA	CB	CC	A	R
1	1	0	1	1	1	0	2
2	1	0	1	0	0	0	1
3	1	0	0	1	1	1	0

4.1 Modelling the terminal

We model the terminal and the container displacements as a network of flow (Papadimitriou and Steiglitz, 1982). Consider, for the moment, the first shift in Tab. 1: a total number of 100 containers must transit from the areas to the ship through the quay cranes at quay 1. The graph in Fig. 3 describes this process.

The square nodes, which are sources of flow, represent the yard areas. The circle node stands for the quay cranes to be assigned to S1 and the rectangular node represents the ship (the sink of flow). The graph represents the paths that containers can follow to reach the ship. Arc labels denote arc capacities, i.e. the maximum number of containers that can flow through an arc during one shift. Arc capacities are proportional to the number of resources allocated for the related node. Deciding which and how many resources to allocate determines the structure of the flow network. This problem is known in literature as network design (Magnanti and Wong, 1984; Ahuja et al., 1993).

Figure 4 reports the flow network extended to all the shifts (arc capacities are not shown).

The structure of the subnetwork over the upper dashed line is the same graph as that in Fig. 1. The nodes are renamed by adding superscripts for shift numbers and subscripts

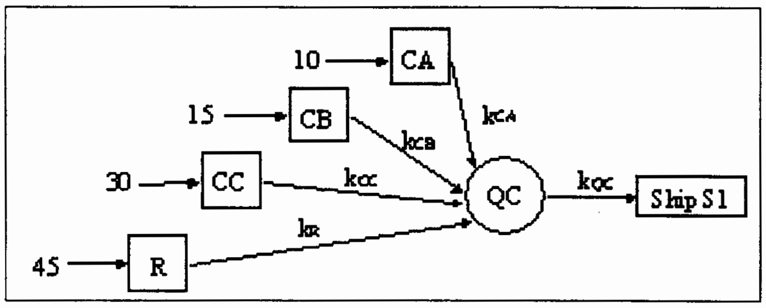

Figure 3: The network of flow for the first shift in Table 1

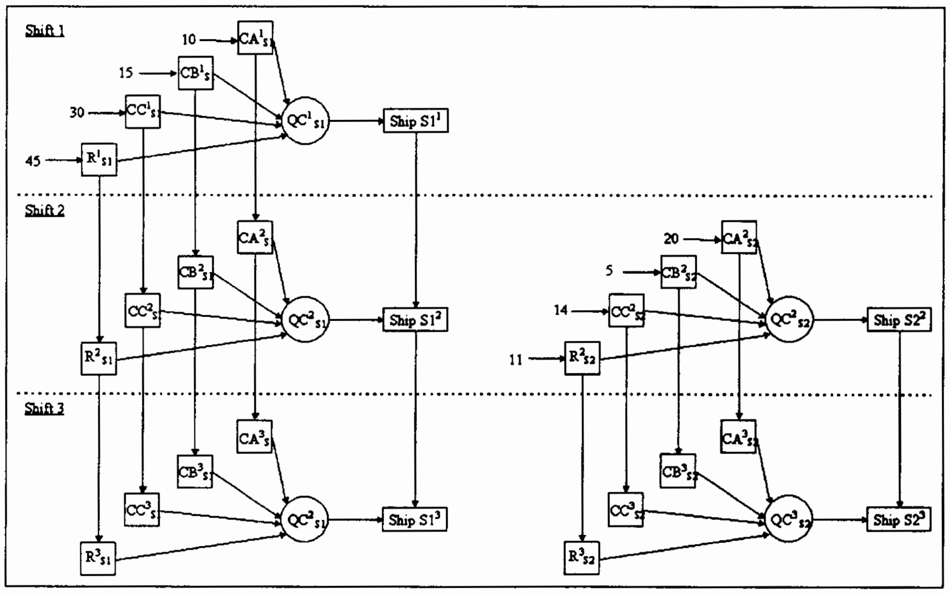

Figure 4: The flow network for the 2 ships described in Table 1

for ship names: e.g., CA^1_{S1} is the subset of containers stored in CA, at the start of shift 1, for ship S1. In this way, we regard each area of the yard as a set of logical subareas related to the shift and to the ship under consideration.

The vertical arcs are temporal arcs that lead to the next shift. For instance, the arc $CA^1_{S1} \longrightarrow CA^2_{S1}$ gives the opportunity of delaying part of the moves of the 10 containers in CA to shift 2. Time arcs are uncapacitated because any number of containers can be postponed to the next shift. Observe that the graph does not extend beyond the third shift, thus forcing the service of S1 to be completed by its deadline.

The presence of two separate graphs in shift 2 is due to the presence of two ships at the terminal; the right side of the graph is used for S2. The two graphs are related by the resources shared between the ships: e.g., the cranes allocated in CA must be shared in order to direct the containers both to QC^2_{S1} and to QC^2_{S2}. This connection is made by using capacity constraints, as detailed in the next section.

4.2 The formal model

By the considerations developed in Section 4.1, we can assume that a graph $G = (V, E)$ and the related (node-arc) incidence matrix $\mathbf{A}$ are available.

Let us define the following structures.

- Let $Y = \{CA, CB, CC, A, R\}$ be the set of the yard areas; let $I = \{1, ..., T\}$ be the set of shifts, T denoting the number of shifts in the period; finally, define $Q = \{q_1, q_2, q_3, q_4\}$ as the set of the quays.
- Let $Y^i \subseteq Y$, be the set of yard areas in the graph at time i; in the same way, let $Q^i \subseteq Q$ be the set of quays represented by nodes of the graph at time i.
- Let $\overline{y}$ be the maximum number of resources available in area $y \in Y$; the maximum number of quay cranes on pier West and pier East are respectively denoted by $\overline{q}_w$ and $\overline{q}_e$. Define the vectors: $\overline{\mathbf{q}}_w = [\overline{q}_w, \ldots, \overline{q}_w]$, $\overline{\mathbf{q}}_e = [\overline{q}_e, \ldots, \overline{q}_e]$ and $\overline{\mathbf{y}}^i = [\overline{y}]_{y \in Y^i}$.
- For each $y \in Y$, we denote by $k_t : \mathbb{N} \rightarrow \mathbb{N}$ the linear function such that $k_t\,(n_r)$ reports the capacity of n_r resources of area (or quay) t (and is undefined if $n_r < 0$ or $n_r > \overline{t}$). In the following this will be taken as the product of the number of allocated resources and the capacity of one resource.

Now some conventions about vectors of variables are introduced for synthesizing the problem.

Consider the variables representing flows. Denote by $f^i_{s,t}$ the variable for the flow to ship s at time i, departing from $t \in Y^i \cup Q^i$. As general convention, whenever the superscript i or some of the subscripts are substituted with (sub)sets, the corresponding (boldface) symbol represents a vector; for instance: $\mathbf{f}^i_{S,Y^i}$ is the vector of all the flows departing from area nodes at time i (where S is the set of the ships). For simplifying the notation, an empty superscript, or subscript, stands for the complete set; hence $\mathbf{f}^i_{S,Y^i}$ can also be written as $\mathbf{f}^i_{Y^i}$. In the same way, $\mathbf{f}$ denotes the vector of all the flow variables. Analogous conventions hold for the allocation variables. For each $i \in I$ and for each $t \in Y^i \cup Q^i$, r^i_t denotes the number of allocated resources of area (or quay) t at shift i; $\mathbf{r}^i_{Y^i}$ stands for the vector of area allocation variables $[r^i_t]_{t \in Y^i}$; $\mathbf{r}^i = [r^i_t]_{t \in Y^i \cup Q^i}$; $\mathbf{r}$ is the vector of all the allocation variables of the model.

- Finally, $\mathbf{c}$ represents the vector of costs related to the allocation of the resources.

The formal model for the port RA problem is:

$$\min \mathbf{cr} \tag{1}$$
$$\mathbf{Af} = \mathbf{f}_0 \tag{2}$$
$$\mathbf{f}_t^i \cdot \mathbf{1} \leq k_t\left(r_t^i\right), \forall i \in I, \forall t \in Y^i \cup Q^i \tag{3}$$
$$\mathbf{r}_{Y^i}^i \leq \overline{\mathbf{y}}^i, \forall i \in I \tag{4}$$
$$(\mathbf{r}_{q_1} + \mathbf{r}_{q_2}) \leq \overline{\mathbf{q}}_w, (\mathbf{r}_{q_3} + \mathbf{r}_{q_4}) \leq \overline{\mathbf{q}}_e \tag{5}$$
$$\mathbf{r}, \mathbf{f} \geq \mathbf{0} \tag{6}$$
$$int\ \mathbf{r}, \mathbf{f} \tag{7}$$

The objective of the problem is the minimization of resource costs. This is made by properly instantiating the elements of $\mathbf{r}$, which, in turn, affect the capacities of the arcs according to the functions k_t. Note that $\mathbf{c}$ allows different costs to be defined for the same resource in different shifts. This is needed because, at the terminal under study, the allocation of the same resource is cheaper during the day as compared to a night allocation.

Constraints (2) express the common *flow conservation law*: the flow that enters a node must also leave the node unless the node is a source or a sink. A generic element of the known vector $\mathbf{f}_0$ is the difference between the incoming and the departing flow. It is non zero if and only if the node is either a source or a sink node.

As far as constraints (3), they express the limits on arc capacities, which depend on the allocated resources. Consider the area nodes (the case of quay cranes is similar). There is one constraint for each combination of shift number and area. The left-hand side of the constraint is the sum of the flows departing from the nodes representing the area in the considered shift. The right-hand side is the total movement capacity of the area.

Constraints (4) and (5) are reported in order to limit the number of available resources, that is the range of values of the elements in $\mathbf{r}$. In particular, both constraints (5) are related to the pier West and pier East, respectively. Consider the one on the left, the other is analogous. Any element of the vector $(\mathbf{r}_{q_1} + \mathbf{r}_{q_2})$ is the number of allocated quay cranes at pier West and at a given shift. This is bounded by the maximum number of quay cranes on pier West, i.e., $\overline{q}_w$. By this bound, we allow any possible partition of the allocated cranes at pier West, between quay 1 and quay 2. This is what happens at the terminal, since the quay cranes can be moved inside the same pier to arbitrarily dimension the movement capacity between the two quays.

Finally, constraints (6) guarantee that the flows and the resources are positive and constraints (7) require that such vectors are integer. The resource variables must be integer because the terminal managers cannot allocate fraction of resources even if resources are only partially used in a given shift (the cost must be paid entirely). Concerning flows, the constraint is needed since the flows represent containers and because they are not generally integer at the optimum without requiring it (Zaffalon and Gambardella, 1998).

5 Implementation

The model given in Section 4.2 is the core of a computational system for the port RA problem. The system was implemented in C. The program takes a 28-shifts workplan as input and outputs an allocation plan for the entire period. It is constituted by a preprocessor module followed by a branch and bound search. The preprocessor modifies the resource capacities as a way to model some special cases. For instance, the container moves by train and trucks, i.e. to containers that are brought into the terminal from outside and vice versa, can be modeled in this way. The branch and bound search is described in the following section.

5.1 Branch and bound search

The core model is solved by a branch and bound search by using Cplex 6 (Ilog, 1998) on a 500-Mhz Pentium with 128 Megabytes of RAM. Exhausting the search tree does not seem viable, so the search is run for a fixed time to get an approximate solution. The quality of the solutions is usually very good, as discussed in Section 6.

There appear to be several reasons for this positive result. The first reason is the preprocessor of Cplex 6. The built-in preprocessor reduced the size (number of rows times number of columns) of the original problem matrix by a factor 0.05 on average. The resulting problem, being much smaller, allowed fast computations to be achieved.

Another reason is related to the search strategy. We found the following search strategy to be particularly effective: explore the tree by depth-first search; choose a branch related to the ceiling of a fractional resource variable first. Thus, for example, if the continuous number of cranes in CA at time i (r^i_{CA}) is 3.5 at a given point of the search, we firstly choose the branch related to $r^i_{CA} \geq 4$. This empirically works much better than choosing the floor branch first. Apparently, it is because the value 3.5 suggests that 3 resources are not sufficient to carry out the job. By this search, the first integer solution is usually found very quickly after a few seconds and is not significantly improved by longer computation times; we used a 15-minutes time limit. Longer times are instead needed by different strategies, like choosing the floor of a fractional variable first, to approach the same value of the former strategy, after a progression of integer solutions.

The search was also improved by noting that the formulation given in Section 4.2 enables the branch-and-bound search to produce naive solutions, when used without exploring the entire tree, which have an evidently excessive number of resources. For example, it is possible that the system allocates $n' > n$ cranes to move the number of containers for which n cranes would suffice. This was corrected by adding the following constraints to the original formulation:

$$k_t\left(r^i_t\right) - \mathbf{f}^i_t \cdot \mathbf{1} \leq k_t\left(1\right) - 1, \forall i \in I, \forall t \in Y^i \cup Q^i. \quad (8)$$

These simply state that the residual capacity (i.e. the difference between the allocated capacity and the moved flows) must be less than the capacity of one resource, namely

$k_t(1)$. The strict inequality is realized by requiring the residual capacity being less than or equal to $k_t(1) - 1$, because the capacities are assumed to be integer.

Finally, we can identify two structural reasons for the good performance of the system. One seems to be the special graph generated by the port RA problem. Although the graph can be quite large, its structure is simple, highlighting that the focus of the problem is on the proper way of *sharing* resources. The second characteristic is concerned with the objective function (1) that does not involve flow variables; so that, once the resource variables are fixed, the remaining flow problem is just a problem of feasibility.

5.2 System parameters

The system for the optimization of the resources is calibrated on the basis of a set of parameters. The movement capacities of the different resources are structural parameters. These are the average number of containers per shift that can be moved by quay cranes, yard cranes and front lifters. These were obtained by consulting domain experts at the LSCT and by using the simulator of the LSCT.

The resource costs, reported in Table 3, are the economic parameters of the system. Note that there are different costs for daily and night shifts, the latter (from 7 pm to 7 am) being more expensive.

Table 3: Unitary resource cost per shift in Euros.

Shift	Cranes			Front lifters
	CA,CB,CC	A	QC	
Daily	199.25	215.67	752.27	231.48
Night	230.86	247.28	830.36	263.08

6 Experimental analysis

This section reports the results of the experimental study carried out for the empirical evaluation of the overall system. The data for the experiment were extracted from the LSCT database that records the movements of each container in the terminal. The data describe the overall activity of the terminal for two consecutive weeks of 1998. The experimental method is based on a loop that involves using the RA module, the scheduling module and the simulator, as reported in Section 3.

Table 4 reports the optimized allocations and compares them with the resources that were actually allocated at the LSCT in the same week.

Table 4: The comparison of the two plans.

Allocations	RA module				LSCT			
	Cranes			Lifters	Cranes			Lifters
	CA+CB+CC	A	QC		CA+CB+CC	A	QC	
Daily	63	14	36	46	96	11	36	63
Night	38	3	14	12	72	2	31	45
Total	101	17	50	58	168	13	67	108
Euros	77,598.45				117,861.57			

The comparison emphasizes the effectiveness of the computed allocations. The overall number of allocated resources is remarkably lower than that of the real allocations. There is a much better partition of the resources between day and night shifts, the former being less expensive, so enabling significant savings. The straight comparison of the total costs, in the last row, shows that the computed plan allows the scheduled containers to be moved by spending only about 67% of the real terminal costs. Also, the RA solution was only 6.6% far, on average, from the continuous bound provided by Cplex 6.

We remark that this good performance has been observed by running the overall system for the port optimization that is based on the interplay among the RA module, the scheduler and the simulator of the LSCT. The allocation policies obtained by the RA module have been shown to be sustainable in the complex stochastic environment provided by the simulator. All the containers were moved within the deadlines of the ships by means of the resources in the RA plan. This emphasizes the realism of the solutions provided by the proposed module.

7 Conclusions

The paper presents a methodological approach to the optimization of an intermodal terminal. The problem of resources allocation and scheduling of containers is solved by using two different but strictly interconnected modules. The resources allocation module remarkably reduces the number of resources typically required by the terminal. The optimized plans tend to exploit the resources to their limit. The scheduler copes with such an increased complexity thanks to its effective solution procedure that allows reducing the conflicts of the yard cranes and, by this, increasing their throughput.

The overall approach is empirically validated by coupling the above modules with a low-level simulator of the port, which shows the effectiveness and the realism of the proposed policies.

Acknowledgments

The authors would like to thank the management group of the La Spezia Container Terminal for their effective collaboration in this research project. Thanks also to Data and

System Planning SA, in Manno (CH), for support in the software development. This project was funded by the Swiss Commission for Technology and Innovation under the project # 3128.1.

References

Aarts E. H., Lenstra J. K. (1997) Introduction. In: *Local Search in Combinatorial Optimization*, Aarts E. H. and Lenstra J. K., eds., Chichester, John Wiley & Sons, 1–17.

Ahuja R., Magnanti T., Orlin J. (1993) *Network Flows*, Prentice Hall, New Jersey.

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

CACI Products Company (1996) *Modsim III, the language for object-oriented programming*, users' manual, La Jolla, CA.

Erard P-J, Déguénon P. (1996) *Simulation par événements discrets*, Lausanne Presses Polytechniques et Universitaires Romandes.

Gambardella L.M., Mastrolilli M., Rizzoli A.E., Zaffalon M. (2001) An optimization methodology for intermodal terminal management. *Journal of Intelligent Manufacturing*, 12, 521–534.

Gambardella L.M., Rizzoli A.E., Zaffalon M. (1998) Simulation and planning of an intermodal container terminal. *Simulation*, 71, 107–116.

Garey M.R., Johnson D.S., Sethi R. (1976) The complexity of flowshop and jobshop scheduling. *Math. Oper. Res.*, 1, 117–129.

Glover F. (1989) Tabu search - Part I. *ORSA Journal on Computing*, 1, 190–206.

Hayuth Y., Pollatschek M.A., Roll Y. (1994) Building a port simulator. *Simulation*, 63, 179–189.

Ilog (1998), *Cplex 6*, www.ilog.com.

Kim H. K., Bae K. H. (1999) Segregating space allocation models for container inventories in port container terminals. *International Journal of Production Economics*, 59, 415–423.

Kim H. K., Kim Y. K. (1999) Routing straddle carriers for the loading operation of containers using a beam search algorithm. *Computers and Industrial Engineering*, 36, 109–136.

Kozan, E. and Preston, P. (1999) Genetic algorithms to schedule container transfers at multimodal terminals. *International Transactions in Operational Research*, 6, 311–329.

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

Mastrolilli M., Gambardella, L.M. (2000) Effective neighborhood functions for the flexible job shop problem. *Journal of Scheduling*, 3, 3–20.

Papadimitriou H., Steiglitz K. (1982) *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, New York.

Sha O.P. (1985) Computer aided on board management. In: Banda P., Kuo C., eds., *Computer Applications in the Automation of Shipyard Operation and Ship Design V*, Elsevier, Amsterdam, 177–187.

Young Y.W., Seok C.Y. (1999) A simulation model for container-terminal operation analysis using an object-oriented approach. *International Journal of Production Economics*, 59, 221–230.

Zaffalon M., Gambardella L.M. (1998) Optimization of resources in an intermodal terminal. *IDSIA technical report n. IDSIA-20-98.*

Zeigler B.P. (1984) *Multifacetted Modelling and Discrete Event Simulation*, Academic Press.

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