# The Logistics Control System for Container Vessels in the Tandem Sea Port

Zbigniew Frąckiewicz, Franciszek Marecki and Robert Bucki



November 2, 2021

# THE LOGISTICS CONTROL SYSTEM FOR CONTAINER VESSELS IN THE TANDEM SEA PORT


## ABSTRACT

**Background:** The paper solves the problem of controlling the handling of container ships in a tandem seaport. The control system for container ships includes determining the route of ships taking into account random disturbances, ship transport time and the time of unloading/loading. Determining the optimal route of ships is an NP problem. The ship service process is scheduled on a weekly basis. However, the tugboats operate in a shift cycle. It is important to designate a warehouse/wharf where containers are to be unloaded/loaded and to determine restrictions. The port consists of a roadway, ships, unloading and loading quays, canals, and warehouses and can have various structures. The port consists of a roadway, ships, unloading and loading quays, canals, and container warehouses and can have various structures. It is assumed that the ships await service in the port roadstead. The time horizon for the allocation is one week.

Ships waiting in the roadstead have priorities depending on the waiting time. It is assumed that the time of transport, unloading/loading of the vessel and economic parameters are given. The optimization of ship service in the port is multi-criteria. There are entry, exit and internal criteria.

**Methods:** The problem of controlling container ships in a tandem seaport was solved by the method of artificial intelligence using a computer program. The optimization criterion was the minimization of vessel service time in both parts of the port, taking into account restrictions.

**Results:** A logistics system for controlling ships in a tandem seaport with averaged data and random events was developed.

**Conclusions:** The paper presents the mathematical model and the logistics algorithm of a tandem port in which the depth of the channel between the ports is important, tugs transporting ships and the type and capacity of warehouses are taken into account.



**Keywords:** logistics, wharf, optimization, roadstead, container ships.

## INTRODUCTION

Some seaports work closely with a nearby river port to form one economic organism known as a port tandem. Such a container tandem port requires specific ship control, the aim of which is to obtain optimal economic results. It includes the following NP problems:

- determining the ship's route through the ports of $P_i$, $(i = 1, \ldots, I)$ – that is the traveling salesman problem;
- solving the "knapsack" problems;
- weekly schedule of vessel handling in the port (transport and transhipment).

Such problems are solved with the use of artificial intelligence.

Handling such a tandem also includes the following problems:

i) steering the tugs (during a shift cycle);
ii) container management in the warehouse (with imported / exported goods) which is featured in other related publications.

A system is understood as a set in which randomness plays an important role (e.g. weather disturbances). A typical transport system is an airplane or a ship. The degree of complexity of the set of deterministic elements usually leads to the use of a systemic method (e.g. statistical). The modern control system determines decisions made by the software (based on many data) taking into account random disruptions (e.g. transport times, exchange rates).

The logistics system for controlling container ships presented in this publication uses known moments of completion of random operations (i.e. the data is deterministic) [Bucki and Marecki 2006]. Other logistics problems are solved in a similar way [Bucki et al. 2012, Bucki et al. 2013, Bucki et al. 2014]. In this way, in the calculations, natural intelligence (associated with a human) and artificial / bit intelligence (associated with a computer) are used [Kaku 2014]. Natural intelligence is based on signs and information intelligence is based on bits.

A container ship is an object designed to transport many export / import goods mainly in containers. It should not go empty because it "earns" when she transports goods. Therefore, the shipowner selects the ship's route through the ports $P_i$ $(i = 1, \ldots, I)$ so that it is optimal, i.e. it brings the maximum profit (under certain constraints). However, there are $I$! routes (via $I$ ports). If the $I$ number is large (e.g. $I > 50$), it takes too long to generate all the routes (even for a computer). Therefore, determining the optimal route of ships is a problem of NP [Papadimitriou 2002] – that is difficult / time-related (even for a computer that quickly processes digital information). There are computational problems that require the generation of all acceptable variations (e.g. travelling salesman, knapsack) to determine the best solution.

There are polynomial / easy (P classes) and exponential / hard (NP classes) problems, and there is no P algorithm that can exactly solve the NP problem. Problems of the NP class can be solved exactly with exponential methods [Harel 1987]:

a) dynamic programming;

b) breakdown and limitations;

c) multi-step programming;

or heuristic algorithms (which do not guarantee optimality).

There is a set of NP-complete problems (knapsack, packing, scheduling, travelling salesman, etc.) to which other NP problems can be reduced and also more difficult than NP-complete. Typically, the computation time depends on the number of $N$ decision variables which are the sequence of NP problems, e.g. the problem of "loading", etc. [Robling-Deenning 1982].

## TANDEM PORTS

Each ship has a base port $P_0$ where it calls for periodic inspections, repairs, etc. In addition, the shipowner determines the (optimal) route of a merchant ship through the ports $P_i$ $i = 1, \dots, I$ where containers are picked up or left (with exported / imported goods). The ports of the $P_i$ are usually composed of ports:

a) outer $R_i$ (marine, e.g. Świnoujście, Cuxhaven);

b) the inner $S_i$ (river, e.g. Szczecin, Hamburg).

Tandem ports have a common roadstead which is located at sea. For this reason, a "smaller" port is built at the entrance / exit of the home port forming a tandem with the home port (e.g. Szczecin-Świnoujście or Hamburg-Cuxhaven).

The main problem with the tandem of ports is the depth of the channel connecting these ports. Such a canal is a two-sided waterway of considerable length (up to 100 km). A deep canal is an expensive investment and one-way traffic makes transportation difficult. Ships, on the other hand, are getting bigger and bigger. Thus, when fully loaded, they could not enter or leave the inner port. An example of this can be, for example, the case of the landing of a container ship on the sandbank of the Suez Canal.

One or several tugs are used to transport ships through the canal between tandem ports, which has an impact on the implementation of transport schedules. The ship cannot follow the channel "alone". The tugs "work" in a daily cycle and the ships in a weekly cycle.

Let us consider the problem of a container ship at port $P_i$ whose draft $H_i$ exceeds the permissible draft $h_i$ of the port tandem channel. Let us denote by $R_i$ the external port and the internal by $S_i$. The ship leaves some of the containers in the port $R_i$ so that it has a draft of less than $h_i$ in the channel. It is assumed that containers intended for the next port $P_j$, $j = 1,\ldots,J$ can be unloaded in the port $R_i$ or the given port $P_i$.

In the case of the port $P_j$ the collection of containers can begin in the port $R_i$. Containers unloaded in the port $R_i$ but these destined for the port of $P_j$, $j = 1,\ldots,J$ will be picked up (by the same vessel) in the port of $R_i$ on the way back.

The problems to be solved at the port $R_i$ and $S_i$ port belong to the NP class, however, the number $N$, ($N = I - i + 1$) is relatively small. Thus, to determine the optimal solution, the review methods can be implemented. First of all, let us consider the backpack problems of the $R_i$ port and then these of the $S_i$ port. Let us assume that the current draft of the ship at sea is given as $H_i$ and the permissible draft in the channel $h_i$ are given. These data allow us to determine the permissible draft of the ship which is the permissible lifting capacity of the "backpack" (in the NP problem).

Containers intended for $P_i$ can be unloaded in the port $R_i$ but the price for the transport of container $c_{i,n}$ will be lower (by the section $R_i \rightarrow S_i$). In addition, containers intended for $P_j$ can be unloaded in $R_i$ if the ship arrives at the roadstead before the agreed date (and is waiting for the tugs). If the containers are destined for the next port $P_j$, $i < j$, then the additional cost of handling the ship $\Delta c_{j,n}$, which reduces the profit of the ship, is taken into account. Thus, the "knapsack" problem is solved in the port $R_i$ for a given ship with the draft limitation $h_i$. Containers to be unloaded in the port $R_i$ for a given ship should be designated in order to respect the draft $h_i$. The criterion takes into account economic effects (prices, exchange rates, etc.).

The containers intended for $P_i$ must be unloaded in the port of $S_i$ but not unloaded in the port of $R_i$. The time of unloading the containers intended for $R_i$ and the time of loading the containers are important while respecting the depth $h_i$.

In this case, a quay (of the port $S_i$) is designated and is characterised by:

a) an appropriate number of vacancies (for containers);
b) "forecast" container unloading times.

Then, the "backpack" problem is solved to load the containers intended for the next port $P_j$ along the ship's route taking into account the "forecast" of container loading times and the permissible immersion depth $h_i$ of the channel (at the same time $h_i$ may depend on the high tides / low tides of the sea). The criterion which is usually adopted is to minimize the time of the ship's reloading.

## PORT MODELS

In general, port models (due to wharfs) can have: the serial, parallel, serial-parallel, and tree or anti-tree structure. The quays are equipped with port basins where ships can wait for the loading of containers. Such models are described in detail in the monograph [Marecki and Marecki 2007, Bucki et al 2010] where parallel programming was considered with a large number of ships. In this study, it was assumed that the number of ships is small.

Let us consider the tandem of $R$ (outer) and $S$ (inner) ports. The $R$ port has a roadstead and the $S$ home port is located inland. These ports are connected by a two-way channel where port $R$ is smaller (has fewer quays) than port $S$. Both ports share a roadway where ships (for $H_i > h_i$ or $H_i < h_i$) wait for tugs. Vessels appear on randomly disrupted dates despite the fact that their service is scheduled on a weekly basis. If the vessel is not serviced during the week (e.g. in the absence of tugs), its priority for service increases. Ships for which: $H_i > h_i$ must leave some of the tugs in port $R$ (so that their draft is less than $h_i$). In port $S$ (regardless of the structure) - for each ship - the NP problem is dealt with ("backpack"). In addition, ship service is scheduled (assuming the availability of tugs) which is also an NP problem because the tugs replace the crew for each working shift.

Let us assume that the $R$ and $S$ ports have a parallel structure with the number of berths: $M < N$ (unloading and loading berths). The time of unloading / loading the containers is shorter if they are placed in the nearest free place. Thus, two sides of the warehouse can be distinguished (e.g. $L$ and $P$). The ship is unloaded on the $L$ side and loaded on the $P$ side. Trains or TIRs can only use one side ($L$ or $P$).

## NP PROBLEM METHODS

IT systems can be intelligent or ordinary (e.g. visitor records for a porter). Ordinary systems should contain domain knowledge (e.g. Archimedes' law), i.e. natural (human) intelligence. Intelligent IT systems contain artificial intelligence as they relate to bits with different physical interpretations. They are associated with the computer as they are related to computational problems that are not exactly solvable in an exact way, e.g. some nonlinear equations, integrals.

Computational problems can be:

a) easy (polynomial) – P (Polynomial);

b) difficult (exponential) – NP (Non Polinomial).

The difficulty of NP problems is that for $N$ decision variables there are generally $2^N$ solution variants. However, with a large $N$ the problem takes too much computational time. Therefore, it is justified to use heuristic algorithms without a guarantee of optimality.

Problems of NP can be solved by the following methods:

a) dynamic programming;

b) breakdown and limitations;

c) multi-stage programming.

These methods help to solve all the problems, e.g. a travelling salesman, calculating the shortest route through $N$ cities - there are $N!$ all variants which gives $2^N$ routes.

Dynamic programming requires generating a complete state network. The state is defined to represent a partial solution (e.g. in the problems of the backpack as well as travelling salesman). This allows us to eliminate states that do not provide the optimal solution.

"Split and Constraints" is a method that uses the anti-tree analogy to designate a point with the maximum height. In this method successive solutions are generated by trajectories (states) but only one solution is remembered – currently the best $R^A$. This method does not require the entire network but only two trajectories: $x$ and $R^A$. Moreover, the computation time is long because all $x$ trajectories are generated.

The multi-stage programming method includes the advantages of the above methods and the state definition that allows us to obtain another solution from the end state. The multi-stage programming method allows the use of various algorithms:

i) exact - generating states in successive stages ($e = 1, \dots, E$);

ii) almost exact - generating states with trajectories ($j = 1, \dots, J!$);

iii) heuristic - no guarantee of optimality;

iv) randomized - random selection of routes;

v) conversational - route selection by the user.

The almost exact algorithm allows us to determine the probability $p(|V^A - V^X|)$ where: $X^A$ - the final state currently the best; $X$ - the state that can be generated.

Artificial intelligence uses the enormous bit processing speed and memory capacity of the computer. Programming is the essence of computer science [Wirth 1976] so we use artificial intelligence in case of:

a) automatic machines (embedded systems) where memory is scarce;

b) computers with large memory.

## THE MULTI-STEP PROGRAMMING METHOD

As it is known, to solve the exact problem of the NP class review methods should be implemented:

a) dynamic programming;
b) division and restrictions (Branch and Bound - B&B);
c) multi-step programming.

In practice, however, heuristic algorithms are used – class P (without a guarantee of optimality) [Robling-Denning 1982].

An overview of these algorithms for the problem of container ships is presented in the multi-step programming method:

a) determining the optimal route through *I* ports characterized by the high *I* requires a long time needed for calculations;
a. solving the problem of handling container ships (on a weekly basis) requires scheduling - NP;
b) "knapsack" problems NP must be solved for each container vessel.

In addition, tug boats should be assigned to meet vessel service schedules (in a shift cycle). Therefore, the logistic IT system is indispensable.

In the multi-step programming method the decision state, the state value, the (permissible) state generation procedure and the rules for eliminating non-prospective states should be defined.

These definitions depend on the problem being solved:

**Definition 1: The decision state *X* is defined to allow for the scheduling of object handling.** The schedule specifies the sequence and the moment of completion of the service of the facilities (e.g. transport or transhipment of ships).

If the port is structured in parallel, state *X* is a matrix with two columns

$$X = [x_{m,n}], m = 1, \dots, M, n = 1, \dots, N \tag{1}$$

The coordinates of this matrix mean:

$x_{m,*}$ – the *m*-th ship which is being transhipped (at a given quay);

$x_{*,n}$ – the deadline for completion of reloading of the *m*-th vessel.

The $x_{m,*}$ coordinates are defined as 0 (when the ship has not been unloaded) but the ship is not reloaded on Sundays (this requires the initial state to be determined in the next week): $x_{m,*}$ (*T*) where: (*T*) – the completion date of the transhipment process of ships.

**Definition 2: The value of the state $V(X)$ depends on the optimization criterion.** For ships, this value is the time, i.e. the maximum coordinate of the second column of state *X*. Thus, the vessel leaves the roadstead at time 0 but may return to the roadstead at time $Q^k$ where: *k* - the route number, that is:

$$Q^k_{m,*} = Q^k + \tau_{m,*}, m = 0,1,..,M \quad (2)$$

where: $\tau$ - the ship transhipment completion date.

This procedure is presented in detail in [Marecki and Marecki].

**Definition 3: Procedure of generating admissible states** – consists in generating the state of the next stage: $e - (X^e)$ on the basis of the previous state ($X^{e-1,i-}$) of stage ($e-1$).

Let us suppose that from the state ($X^{e-1,i}$) the acceptable state of the next step *e* (i.e. $X^e$) is generated. Therefore, there must be at least as many containers on board as those missing from the warehouse. The procedure of generating admissible states has the $G\ (X)$ limitation in the form:

$$\left(X^{e-1}_{m,*} = 0\right) => \left[X^e = F\left(X^{e-1,i}, m\right)\right] \quad (3)$$

The function *F* determines the permissible transition from the state $X^{e-1,i}$ to the state $X^e$ that is:

$$min\left[V\left(X^{e-1,i}\right) + \tau_m\right] = V^{e,j} V(X^{0,1}) = 0; \quad (4)$$

**Definition 4: Rule for the elimination of non-perspective states** – requires the creation of mathematical theorems to eliminate a state which does not give the optimum or a feasible solution.

This rule may result from dynamic programming. For example, a vehicle can go through cities (e.g. A, B and C) along 6 different routes. Only the state with the shortest time is optimal. The rule for eliminating non-perspective states may result from the B&B method or signal that the generation of the end state $X^E$ is not possible.

Therefore, the multi-step programming method (the solution to the NP class problem) consists in carrying out calculations in the network of states with: $E + 1$ stages and $J_e$ states within the *e*-th stage ($e = 0,1, \ldots, e, \ldots, E$). The only state of the $e = 0$ stage is the given initial state while the best state of the final stage is the optimal solution to the problem.

In practice, the number of containers in the warehouse may vary from day to day. Then another quay is designated for the *m*-th ship.

## THE OPTIMAL MULTI-STEP PROGRAMMING ALGORITHM

The multi-stage programming algorithm with the FIFO rule allows us to generate the optimal solution (based on the states of the last stage) so it is not necessary to remember the entire network of states but only the states of two adjacent stages. The first state generated is the first "erased" from the main memory. This method does not require memorizing the entire network, but only the states of two neighbouring stages: $< 0;\ 1 >, \ldots, < e-1;\ e >, \ldots, < E-1;\ E >$. Steps $e = 0$ and $e = E$ have only one state (initial $X^0$ and optimal $X^{\mathrm{OPTIMUM}}$). From step $e = 0$, the permissible states can be generated: $X^{1,1}$,..., $X^{1,j}$, ... , $X^{1,J}$. Theoretically $J = M$ but in practice $J < M$. Similarly, from the stage $E - 1$ we get only one state $X^E = X^A = X^{OPTIMAL}$. From this state, the optimal solution is obtaimed. However, most states are obtained for $e = E / 2$. If there are too many states, the almost optimal solution can be obtained (taking the best states). Applying (from $e = 2$) the rule of elimination of non-prospective states, a significant reduction of the network is obtained. For example, a ship can "pass" through $P_i$ ports (e.g. A, B and C) along 6 different routes. However, only the shortest time state is the best. The rule for eliminating non-perspective states may also signal that the generation of the state $X^{e,j}$ does not lead to the end state of $X^E$. Thus, the multi-stage programming method with the FIFO rule allows us to determine only the optimal solution as the best state of the last stage.

## CONCLUSIVE REMARKS

The paper solves the problem of handling container ships in a tandem port, i.e. external with the roadstead ($R_i$) and the internal one inland ($S_i$) connected by a two-way channel with limited draft. The operated objects are ships. The control consists in allocating ships to quays. There are transport and ship loading / unloading operations at quays. It is assumed that the ships await service in the port roadstead. The time horizon for the allocation is one week. Ships waiting in the roadstead have their priorities changed depending on the waiting time in the roadstead. The optimization of ship service in the port is multi-criteria. The optimization criterion is minimizing the ship's waiting time in the roadstead, minimizing the cost of quay downtime or maximizing the economic effects of the port. The mathematical models and algorithms of the container port logistics are presented.

## REFERENCES

Bucki. R., Marecki, F., 2006, Digital Simulation of Discrete. Network Integrators Associates, Parkland, Florida, 162, ISBN 978-09788606-7-7.

Bucki, R. et al. 2012, Information Logistics. Institute of Management and Information Technology. Bielsko-Biała, 77, ISBN 978-83-62466-17-7.

Bucki, R. et al. 2013, Proceedings of the International Workshop: Information Logistics, Institute of Management and Information Technology, Bielsko-Biała, 25-26 October 2013, 159, ISBN 978-83-62466-21-4.

Bucki, R. et al. 2014, Information Logistics, Institute of Management and Information Technology, Bielsko-Biała, 110, ISBN 978-83-62466-48-1.

Bucki, R., Frąckiewicz, Z., Marecki, F, 2010, The Information System for Servicing Ships in the Port, Optimization and Simulation Methods, Project: Information Logistics of Transport, Production and Storing Systems, Id. No. MEB051024. Faculty of Applied Informatics, Tomas Bata University in Zlín, 10-12 November 2010, 43-47, ISBN 978-80-7318-989-1.

Harel, D. Algorithmics, 1987, The Spirit of Computing. The Weizmann Institute of Science, Rehovot, 428, ISBN 83-204-2688-X.

Kaku, M, 2014, The Future of the Mind: The Scientific Quest to Understand, Enhance, and Empower the Mind. Doubleday, New York, 400, ISBN 978-03-855-308-28.

Marecki, J., Marecki, F., 2007, Service of Ships at Ports, Information Logistics, Institute of Management and Information Technology, Bielsko-Biała, 104, ISBN 978-83-607-165-40.

Papadimitriou, Ch. H., 2002, Computational Complexity. Addison Wesley Longman, 540, ISBN 83-204-2659-6.

Robling-Denning, D. E., 1982, Cryptography and Data Security. Addison Wesley Publishing Company, 427, ISBN 83-204-1658-2.

Wirth, N., 1976, Algoritms + Data Structure = Programs. ETH, Zurich, 384, ISBN 83-204-2740-1.