On a certain berth scheduling problem

Jacek Błażewicz, Maciej Machowiak ¹  T.C. Edwin Cheng ²  Ceyda Oğuz ³

1 Introduction

Berth allocation problem is to determine the berthing time and the position of each ship along the quayside. This is among the most important decision problems in a port container terminal since a good allocation of berths to the incoming ships will enhance ship owners’ satisfaction and increase terminal productivity, leading to higher revenues.

We model the berth allocation problem (see, for example, [6, 7, 10, 11]) as a moldable task scheduling problem by considering the relation between the number of quay cranes and the berthing time. Moldable tasks form one type of parallel tasks that can be processed simultaneously on a number of parallel processors for which the processing times are a function of the number of processors assigned. The number of processors assigned to the moldable tasks is determined at the time of scheduling and will not be changed until the tasks complete their processing. In our model the quay cranes arranged along the berths are the processors and ships are the tasks to be processed by consecutive processors without preemptions. The aim in this model is to minimize the idle time on processors so as to increase the utilization of the berths. The necessity of minimizing turn-around time of the ships for both the port operators and the ship owners establishes the validity of this objective. From the moldable task scheduling point of view, this will be achieved by minimizing the maximum completion time of all tasks, that is the schedule length. The moldable task scheduling model was first proposed by [12] and later studied by [8] and [9].

2 Problem formulation

Our approach is to consider the berth allocation problem as a non-preemptable moldable task scheduling problem. This approach can simultaneously increase the utilization of quay cranes, shorten the turn-around time of ships, and decrease the waiting time of the containers.

The moldable task scheduling problem can be stated formally as follows: we consider a set of \( m \) identical processors (quay cranes) used for executing the set \( T \) of \( n \) independent, non-preemptable (i.e. each task is processed by a constant number of processors from its start to completion) moldable tasks (ships). Each moldable task (MT) has to be executed for the period

\[ \text{1 Introduction} \]


erth allocation problem is to determine the berthing time and the position of each ship along the quayside. This is among the most important decision problems in a port container terminal since a good allocation of berths to the incoming ships will enhance ship owners’ satisfaction and increase terminal productivity, leading to higher revenues.

We model the berth allocation problem (see, for example, [6, 7, 10, 11]) as a moldable task scheduling problem by considering the relation between the number of quay cranes and the berthing time. Moldable tasks form one type of parallel tasks that can be processed simultaneously on a number of parallel processors for which the processing times are a function of the number of processors assigned. The number of processors assigned to the moldable tasks is determined at the time of scheduling and will not be changed until the tasks complete their processing. In our model the quay cranes arranged along the berths are the processors and ships are the tasks to be processed by consecutive processors without preemptions. The aim in this model is to minimize the idle time on processors so as to increase the utilization of the berths. The necessity of minimizing turn-around time of the ships for both the port operators and the ship owners establishes the validity of this objective. From the moldable task scheduling point of view, this will be achieved by minimizing the maximum completion time of all tasks, that is the schedule length. The moldable task scheduling model was first proposed by [12] and later studied by [8] and [9].

\[ \text{2 Problem formulation} \]

Our approach is to consider the berth allocation problem as a non-preemptable moldable task scheduling problem. This approach can simultaneously increase the utilization of quay cranes, shorten the turn-around time of ships, and decrease the waiting time of the containers.

The moldable task scheduling problem can be stated formally as follows: we consider a set of \( m \) identical processors (quay cranes) used for executing the set \( T \) of \( n \) independent, non-preemptable (i.e. each task is processed by a constant number of processors from its start to completion) moldable tasks (ships). Each moldable task (MT) has to be executed for the period

\[ \text{1 Introduction} \]


erth allocation problem is to determine the berthing time and the position of each ship along the quayside. This is among the most important decision problems in a port container terminal since a good allocation of berths to the incoming ships will enhance ship owners’ satisfaction and increase terminal productivity, leading to higher revenues.

We model the berth allocation problem (see, for example, [6, 7, 10, 11]) as a moldable task scheduling problem by considering the relation between the number of quay cranes and the berthing time. Moldable tasks form one type of parallel tasks that can be processed simultaneously on a number of parallel processors for which the processing times are a function of the number of processors assigned. The number of processors assigned to the moldable tasks is determined at the time of scheduling and will not be changed until the tasks complete their processing. In our model the quay cranes arranged along the berths are the processors and ships are the tasks to be processed by consecutive processors without preemptions. The aim in this model is to minimize the idle time on processors so as to increase the utilization of the berths. The necessity of minimizing turn-around time of the ships for both the port operators and the ship owners establishes the validity of this objective. From the moldable task scheduling point of view, this will be achieved by minimizing the maximum completion time of all tasks, that is the schedule length. The moldable task scheduling model was first proposed by [12] and later studied by [8] and [9].

\[ \text{2 Problem formulation} \]

Our approach is to consider the berth allocation problem as a non-preemptable moldable task scheduling problem. This approach can simultaneously increase the utilization of quay cranes, shorten the turn-around time of ships, and decrease the waiting time of the containers.

The moldable task scheduling problem can be stated formally as follows: we consider a set of \( m \) identical processors (quay cranes) used for executing the set \( T \) of \( n \) independent, non-preemptable (i.e. each task is processed by a constant number of processors from its start to completion) moldable tasks (ships). Each moldable task (MT) has to be executed for the period

\[ \text{1 Introduction} \]


erth allocation problem is to determine the berthing time and the position of each ship along the quayside. This is among the most important decision problems in a port container terminal since a good allocation of berths to the incoming ships will enhance ship owners’ satisfaction and increase terminal productivity, leading to higher revenues.

We model the berth allocation problem (see, for example, [6, 7, 10, 11]) as a moldable task scheduling problem by considering the relation between the number of quay cranes and the berthing time. Moldable tasks form one type of parallel tasks that can be processed simultaneously on a number of parallel processors for which the processing times are a function of the number of processors assigned. The number of processors assigned to the moldable tasks is determined at the time of scheduling and will not be changed until the tasks complete their processing. In our model the quay cranes arranged along the berths are the processors and ships are the tasks to be processed by consecutive processors without preemptions. The aim in this model is to minimize the idle time on processors so as to increase the utilization of the berths. The necessity of minimizing turn-around time of the ships for both the port operators and the ship owners establishes the validity of this objective. From the moldable task scheduling point of view, this will be achieved by minimizing the maximum completion time of all tasks, that is the schedule length. The moldable task scheduling model was first proposed by [12] and later studied by [8] and [9].

\[ \text{2 Problem formulation} \]

Our approach is to consider the berth allocation problem as a non-preemptable moldable task scheduling problem. This approach can simultaneously increase the utilization of quay cranes, shorten the turn-around time of ships, and decrease the waiting time of the containers.

The moldable task scheduling problem can be stated formally as follows: we consider a set of \( m \) identical processors (quay cranes) used for executing the set \( T \) of \( n \) independent, non-preemptable (i.e. each task is processed by a constant number of processors from its start to completion) moldable tasks (ships). Each moldable task (MT) has to be executed for the period
of \( t_i(r) \) (processing time) on \( r \) processors, where \( 1 \leq r \leq m \). The number of processors allotted to a task is unknown in advance. The above property distinguishes moldable tasks from the multiprocessor tasks (size model), considered for example in [1], where the number of processors allotted to each task is known. Also this property differentiates moldable tasks from malleable tasks (see, for example, [2]) for which the number of processors allotted to a task is also unknown but may change during their execution. As a result, the processing speed of a moldable task depends on the number of processors allotted to it. We let \( f_i(r) \) denote the processing speed function for task \( i \) if \( r \) processors are used to execute task \( i \). The dependence of a moldable task processing time on a number of processors allotted is given as a discrete function; that is, it takes values at the integer points only. The criterion assumed is the schedule length, which is denoted by \( C_{\text{max}} \) with \( C_{\text{max}} = \max \{ C_i \} \), where \( C_i \) denotes the completion time of task \( i \).

The problem of scheduling independent MT without preemption is NP-hard ([4]), thus, suboptimal algorithms were looked for. A survey of moldable tasks with discrete processing time functions is given by [3] and general moldable task scheduling problems with precedence constraints are reviewed by [5].

From a different point of view, we may consider a problem in which the processors represent a continuously divisible renewable resource bounded from above ([13]). In this problem the processor (resource) allocation is not required to be integer. In [2] it was shown that an optimal solution to such a problem in which processing speed functions are interpolated by piecewise linear functions between integer points (P-CNTN problem) can be found in polynomial time, assuming that the processing speed functions are all concave. In what follows, we will refer the optimal schedule length for such a solution as \( C^*_{\text{cont}} \) and will use it to construct a suboptimal solution for our discrete berth scheduling problem in question. (The procedure for solving the continuous version of the problem will not be discussed here. We refer the readers to the relevant papers by [13] and [2]).

### 3 Suboptimal algorithm and computational experiments

We propose a suboptimal algorithm to solve the moldable task scheduling problem, which was used above to model the berth allocation problem. The proposed algorithm constructs schedules for concave discrete processing speed functions with a very good average behavior. It starts from the continuous version of the problem (P-CNTN, i.e. with a piecewise linear interpolation of the discrete function) and transforms the schedule obtained from the continuous version into a feasible schedule for the discrete MT model. In the proposed algorithm, a rounding scheme for a non-integer allocation is firstly described. Next, the tasks are thoroughly packed using several steps of rounding off. These steps allow for a good average behavior, as is demonstrated in the computational experiments, for a wide range of task and processor parameters.

To evaluate the average behavior of the Algorithm we used the following measure:

\[
\text{Relative error} = \frac{C'_{\text{max}}}{C^*_{\text{cont}}}
\]

where \( C'_{\text{max}} \) is the schedule length obtained by the Algorithm and \( C^*_{\text{cont}} \) is the optimal schedule length of the continuous solution.

Clearly, the value \( C^*_{\text{cont}} \) is the lower bound on the optimal schedule length \( C^*_{\text{max}} \) for moldable tasks, thus, \( \text{Relative error} \) indicates properly the behavior of the Algorithm.
Task processing times $t_i(1)$ have been generated from a uniform distribution in the interval [1..100]. Processing speed functions have been chosen as $f_i(r) = r^a, 0 \leq a \leq 1$. For $n$ equal 16, we varied the value of $m$ between 2 and 48 to see the effect of changing a number of processors on the average performance of the Algorithm under different processing speed functions. Then for $m$ equal to 8, we varied $n$ between 2 and 32 to see the influence of changing a number of tasks on the average performance of the Algorithm under different processing speed functions, (see Figure 1).

![Figure 1: An influence of different shapes of processing speed functions on relative error when a number of tasks is constant (n=16, upper figure) or a number of processors is constant (m=8, lower figure).](image)

References


