Exact and heuristic methods to solve the berth allocation problem in bulk ports

Nitish Umang [*] Michel Bierlaire [*] Ilaria Vacca [*]

August 15, 2012

Report TRANSP-OR 120617
Transport and Mobility Laboratory
Ecole Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

[*] Transport and Mobility Laboratory (TRANSP-OR), School of Architecture, Civil and Environmental Engineering (ENAC), Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland, {nitish.umang, michel.bierlaire, ilaria.vacca}@epfl.ch

Abstract

The Berth Allocation Problem (BAP) is one of the most critical and widely studied problems in port operations. While significant contributions have been made in the use of operations research methods and techniques to solve the BAP in container terminals, almost no attention has been directed to bulk ports. In this paper, we study the berth allocation problem in bulk ports for hybrid berth layout and dynamic vessel arrivals. A key difference that distinguishes the berth allocation problem in bulk ports from that in container terminals is that it is necessary to account for the cargo type on the vessel. In our model, the cargo locations on the yard and the locations of the fixed facilities such as conveyors and pipelines along the quay are explicitly taken into consideration in modeling the handling times of the vessels berthing at the port. The objective of the allocation is to minimize the total service time of all vessels berthing at the port in a given planning horizon. For a given yard layout of the bulk terminal and given locations of fixed facilities such as conveyors and pipelines along the quay, our model enhances coordination between berthing and yard activities. We present a mixed integer linear programming (MILP) approach to model the problem, and an alternate exact solution approach based on generalized set partitioning. A heuristic approach based on the principle of squeaky wheel optimization is also presented. We compare the formulations from a computational perspective through extensive numerical experiments based on instances inspired from real data obtained from SAQR port, Ras Al Khaimah, UAE, the biggest bulk port in the middle east. Our research problem derives from the realistic requirements of the port where currently the waiting times for the vessels are very large. The results indicate that the set partitioning approach and the heuristic approach can be used to obtain near-optimal solutions for even larger problem size.

1 Introduction

The Berth Allocation Problem (BAP) refers to the problem of assigning a set of vessels to a given berth layout within a given time horizon. There could be several objectives such as minimization of the service times to vessels, minimization of port stay time, minimization of number of rejected vessels, minimization of deviation between actual and planned berthing schedules etc. There are several spatial and temporal constraints involved in the BAP, which lead to a multitude of BAP formulations. The existing models for BAP in literature can be classified on the basis of the temporal attributes such as vessel arrival process, start of service and handling times of vessels as well as spatial attributes relating to the berth layout, draft restrictions and others.

According to Bierwirth and Meisel (2010), the vessel arrival process can be considered as static or dynamic. In the static case, the arrival times do not impose a hard constraint on the berthing times, and vessels can berth at any time given that the allocated portion of the quay is available for berthing. In the dynamic variant of the problem, vessels cannot berth before they have arrived at the port. The vessel arrivals can be deterministic in which fixed expected values of arrival times are given, or stochastic in which a distribution of arrival times may be given to account for uncertainty in vessel arrival times.

The handling times for vessels can be assumed as fixed and unchangeable, or dependent on the berthing positions of vessels and/ or work schedule and number of cranes assigned to vessels. The handling times may also be considered as stochastic to account for uncertainty in handling times due to unforeseen disruptions such as equipment breakdown or unavailability of equipment or cargo due to any other reason.

Spatial constraints limit the feasible berthing positions of vessels according to a preset partitioning of the quay into berths. On the basis of berth layout, the BAP can be classified as discrete, continuous or hybrid (Bierwirth and Meisel (2010)). In the discrete case, the quay is divided into a set of sections or berths, and a given berth can be used by at most one vessel at any given time. In the continuous case, there is no partitioning of the quay, and a vessel can occupy any arbitrary position along the quay. This understandably leads to better utilization of the quay space, but is computationally more complicated. In the hybrid case, the quay is partitioned into a set of sections, but a vessel can occupy more than one section at a time, and more than one vessel may be allowed to share the same section at the same time. A graphical representation of different berth layouts is shown in Figure 1. In addition, the draft restrictions on vessels which limit the feasible berthing positions of vessels to only those berths or sections which have a draft higher than the draft of the vessel may also be considered in formulating the BAP.

A feasible berthing assignment can be represented on a space-time graph as shown in Figure 2. The vertical axis corresponds to the quay space within the quay boundary, while

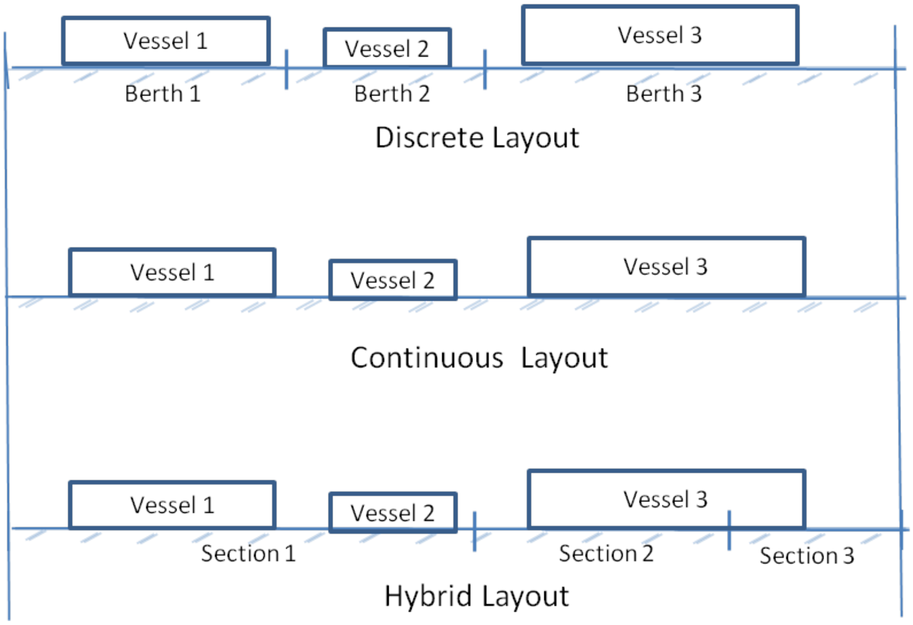

Figure 1: *Berthing Layouts*

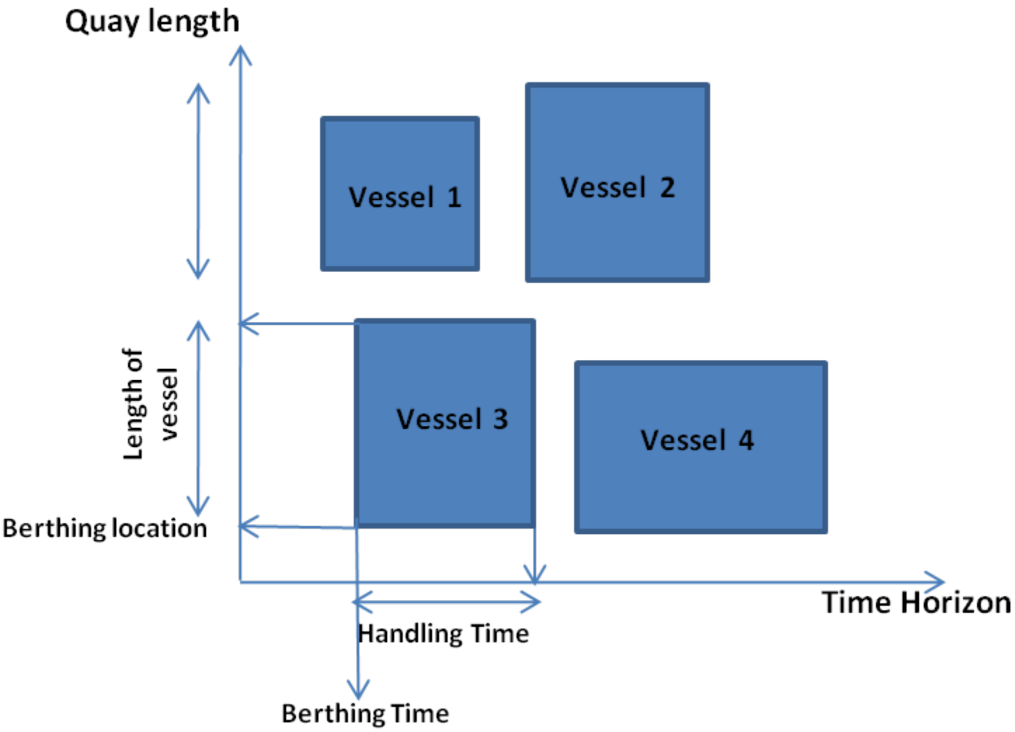

Figure 2: *Feasible BAP Solution.*

the horizontal axis represents berthing time within the planning horizon. Each rectangle represents a vessel berthing at the port. The height of the rectangle represents the length of the vessel, with the upper and lower co-ordinates indicating the berthing location along the quay. The width of the rectangle represents the handling or processing time of the vessel, with the left and right co-ordinates indicating the start and end of handling time respectively. While two vessels may be overlapping in space or in time, it is infeasible for two vessels to overlap in both space and time simultaneously. Thus, in a feasible berthing assignment, all rectangles (vessels) should be non-overlapping and each individual vessel should respect the spatial and temporal constraints on its berthing. This representation of the BAP on the time-space graph further makes it possible for scholars to study the berth allocation problem as a 2-D bin packing problem (Lim (1998)).

In this paper, we discuss the dynamic, hybrid berth allocation problem in bulk ports. To our knowledge, this is the first paper that studies the berth allocation problem in the context of bulk ports and discusses BAP formulations that explicitly take into account the cargo type on the vessel. A mixed integer linear programming approach is presented to solve the problem. An alternate exact solution algorithm based on generalized set partitioning approach is presented to solve larger and more complex instances of the BAP under study. To solve the problem in large scale realistic environments, a heuristic algorithm based on an optimization approach, commonly termed in literature as squeaky wheel optimization is also developed (Clements et al. (1997)). This technique has been successfully applied in graph coloring and scheduling problems, and more recently used to solve operations research problems in container terminals by few scholars such as Fu et al. (2007) and Meisel and Bierwirth (2008). The approach is adapted to solve the hybrid, dynamic BAP in bulk ports. Numerical experiments are conducted on instances based on real port data to test and validate the efficiency of the proposed algorithms.

2 Problem Definition

We consider a set of vessels N, to be berthed on a continuous quay of length L for a time horizon H. We consider dynamic vessel arrival process and a berth layout which is an extension of the hybrid case. We discretize the quay boundary into a set M of sections of variable lengths. In a feasible berthing assignment, a given vessel may occupy more than one section, however a given section cannot be occupied by more than one vessel or part of a vessel at any given time. Partitioning the quay space into sections of variable length brings more flexibility to the model, and the manner in which sections are defined along the quay is critical.

One major difference that distinguishes the Berth Allocation Problem (BAP) in bulk ports from that in container terminals is the presence of fixed specialized equipment facilities such as conveyors and pipelines at bulk ports. In a container terminal, all cargo is packed into containers, and thus there is no need for any specialized equipment to handle any particular type of cargo. In contrast in bulk ports, depending on the vessel requirements and cargo properties, a wide variety of equipment is used for discharging or loading operations. For example, liquid bulk is generally discharged using pipelines which are installed at only certain sections along the quay. Similarly, a vessel may require the conveyor facility to load cargo from a nearby factory outlet to the vessel. For a given vessel, the handling time has a variable component as determined by the berthing position of the vessel along the quay and a fixed component determined by the number of quay cranes operating on the vessel. The berthing position of the vessel along the quay, determines the distance between the berthing position and the storage location of the cargo type of the vessel on the yard. This in turn determines the time taken to transfer cargo between the berthing location and the cargo location on the yard using auxilliary equipment facilities such as loading shovels, trucks etc. or specialized facilities such as conveyors and pipelines.

We define a single variable co-ordinate system along the quay, with the origin at the left extreme of the quay. The vessels berth from the beginning of the first occupied section. This is schematically shown in Figure 3 for |N| = 3 and |M| = 6.

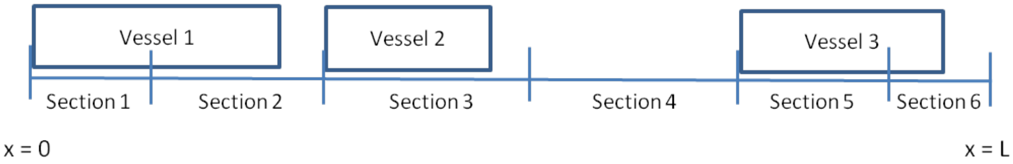

Figure 3: *Schematic representation of vessels berthing along quay of length* L *for* |N|=3*,* |M|=6

To model the hybrid, dynamic berth allocation problem (BAP) in bulk ports, we assume the following input data to be available:

$N =$ set of vessels
$M =$ set of sections
$k =$ 1,...,$|M|$ sections along the quay
$i =$ 1,...,$|N|$ vessels berthing at the port
$A_i =$ expected arrival time of vessel i
$D_i =$ draft of vessel i
$L_i =$ length of vessel i
$Q_i =$ quantity of cargo for vessel i
$W_i =$ set of cargo type(s) to be loaded or discharged from vessel i indexed from w=1 to w=$|W_i|$
$d_k =$ draft of section k
$\ell_k =$ length of section k
$b_k =$ starting coordinate of section k
$h_k^w =$ handling time for unit quantity of cargo type w for vessel berthed at section k
$L =$ total length of quay

The clearance distances between adjacent vessels as well as end-clearances may be considered implicitly in vessel lengths. Similarly, the clearance times between two successive vessels overlapping in space may be considered implicitly in the handling times

In the computation of handling times, the main assumption is that all sections occupied by the berthed vessel are being operated simultaneously with each section handling the amount of cargo proportionally to the section length. The handling time of the vessel is the time taken to load or discharge the section whose operation finishes last. The unit handling time h_k^w for section k and cargo type w includes the time taken to transfer unit quantity of cargo between the cargo location on the yard and the berthed section, and the time taken to load (or unload) the cargo from the quay side to the vessel. In equation 1, these have been denoted by β_k and α_k respectively. Thus we have,

$$h_k^w = \alpha_k^w + \beta_k^w \tag{1}$$

$$\alpha_k^w = T/n_k^w \tag{2}$$

$$\beta_k^w = v^w d_k^w \tag{3}$$

In equation 2, T is the crane handling rate for loading or discharging operations, and n_k^w is the number of cranes operating in section k for cargo type w. β_k^w is the time taken to transfer a unit quanity of cargo between the location of cargo type w on the yard and the section k, which is assumed to be a linear function of the distance d_k^w between the two locations. The parameter v^w depends on the rate of transfer of cargo type w. A schematic representation of a bulk port terminal is shown in Figure 4. As shown, for a vessel carrying cement(w=8) berthed at section k=5, the unit handling time value is $h_5^8 = \alpha_5^8 + \beta_5^8 = T/n_5^8 + \nu^8\ d_5^8$, where d_5^8 is the distance between the section k=5 and the cargo location w=8, ν^8 is a function of the rate of transfer of cement from the cargo location to the berthed section and n_5^8 is the number of cranes operating in the section. Alternatively, if a vessel is using the conveyor facility to load rock aggregates from the rock factory directly into the vessel, the vessel must occupy section k=4. If no additional cranes are being used to transfer the cargo at k=4, we provide $\alpha_4^{11} \longrightarrow 0$, and $h_4^{11} = \beta_4^{11} = \nu^{11}\ d_4^{11}$, where ν^{11} is a parameter dependent on the rate of material transfer in the conveyor and d_4^{11} is the distance between the rock factory and the conveyor facility.

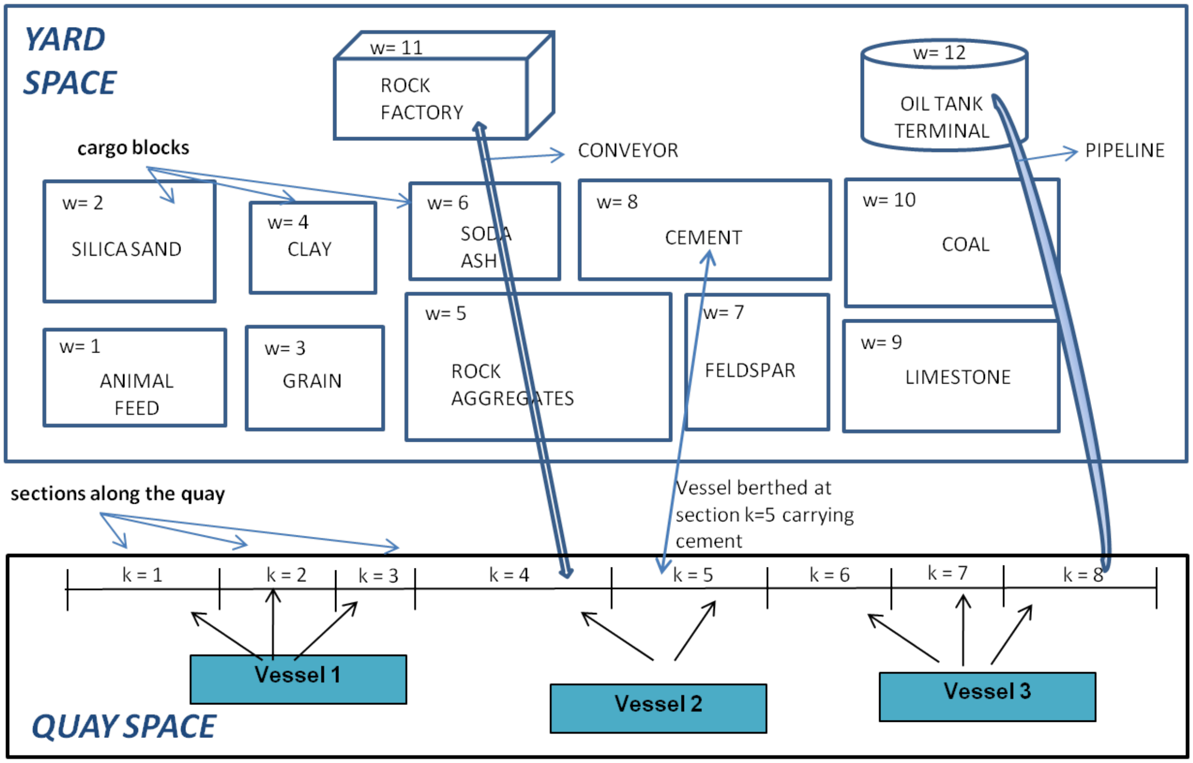

Figure 4: *Schematic representation of a bulk port terminal*

In our model, the specialized equipment facilities such as conveyors and pipelines are also modeled as cargo types. The modeling of handling times using specialized facilities is taken into account by generating appropriate values of unit handling times h_k^w for every section k that may or may not have the cargo facility w. For example, if a particular cargo type w corresponding to the conveyor facility cannot be handled by section k which does not have this facility, the unit handling time value h_k^w is set to infinity or an extremely large value, for this particular section and cargo facility. This prevents the vessels needing the conveyor facility from berthing at any section(s) other than the section(s) where the facility is installed.

3 Literature Review

In this section we present a brief review of past literature on the berth allocation problem in the context of container terminals.

Discrete BAP. The static variant of discrete BAP has been studied by Imai et al. (1997) which minimizes the total service times of vessels and the deviation between arrival order and service order of vessels, Imai et al. (2001) and Imai et al. (2008)). The dynamic discrete BAP problem is considered by Imai et al. (2001), Imai et al. (2003), Monaco and Sammarra (2007), Buhrkal et al. (2011) and few others. More recent approaches, such as Zhou and Kang (2008) and Han et al. (2010), solve the problem considering stochasticity in both arrival times and handling times of vessels. Cordeau et al. (2005) uses a tabu search method to solve the discrete dynamic BAP with due dates, which is further improved upon by Mauri et al. (2008) using a column generation approach that delivers higher quality solutions in lesser computation time. Vacca (2011) study the discrete dynamic BAP at the tactical level in integration with the quay crane scheduling problem, and propose a two-level heuristic to solve the problem.

Continuous BAP. The static continuous BAP has been considered by Li et al. (1998), Guan et al. (2002) and Park and Kim (2003). Guan and Cheung (2004) consider continuous dynamic BAP with fixed handling times using a tree search procedure to minimize the total weighted port stay time of vessels. Gao et al. (2010) use a robust planning approach to solve a dynamic continuous BAP with stochastic vessel arrivals via feedback procedure in the planning stage. Minimization of tardiness as an objective in continuous dynamic BAP is considered by Park and Kim (2002) using a sub-gradient method and by Kim and Moon (2003) using simulated annealing approach. Minimization of quay length with given berthing times as an objective is studied by Lim (1998) and Tong et al. (1999). The continuous BAP with handling times depending on berthing positions is studied by Imai et al. (2005) and Chang et al. (2008) who further considers draft restrictions in the BAP model.

Hybrid BAP. The dynamic hybrid BAP with fixed handling times is considered by Moorthy and Teo (2006), which considers a robust planning approach by incorporating stochasticity in vessel arrivals, and further studied by Dai et al. (2008). The dynamic hybrid BAP with position-dependent handling times is studied by Imai et al. (2007) for indented berths, and Cordeau et al. (2005). Draft restrictions in dynamic hybrid BAP are considered by Nishimura et al. (2001) and Cheong et al. (2010).

Comprehensive literature surveys on the BAP in context of container terminal operations can be found in Bierwirth and Meisel (2010), Steenken et al. (2004) and Stahlbock and Voss (2008). To our knowledge, the problem has not been investigated thus far in the context of bulk port terminals, which is the primary focus of the research presented in this paper. Unlike container terminals, in bulk ports it is necessary to account for the cargo type on the vessel and model the interaction between the yard layout concerning the location of specific cargo types on the yard and the berthing locations of the vessels. Moreover, in container terminals the loading and unloading operations are usually carried out using rail mounted gantry (RMG) cranes that move along a guided rail and cannot pass each other. On the other hand in bulk terminals there is a wide range of heterogenous loading/unloading equipment. This includes specialized equipment facilities such as conveyors and pipelines that are installed at only certain sections along the quay, and mobile harbor cranes that can be freely moved around and can pass each other during the service of a vessel. These differences among others necessitate the need to devise specific solutions for bulk ports. This paper discusses exact and heuristic algorithms to solve the berth allocation problem in context of bulk ports and the algorithms are compared from a computational perspective based on instances inspired from real bulk port data.

4 Models for BAP

In this section, we present two alternate exact solution approaches and a heuristic approach to model the berth allocation problem in bulk ports with hybrid layout and dynamic vessel arrivals. In section 4.1, we describe a mixed integer linear programming approach to solve the problem, while in section 4.2, we use a set partioning approach to model the same problem by apriori generating the set of all feasible berthing assignments by data pre-processing. In section 4.3, we describe a heuristic approach based on squeaky wheel optimization to obtain near-optimal solutions for large instances.

However before proceeding to describing the formulations in detail, we present a mixed integer formulation developed by Nishimura et al. (2001) to solve the berth allocation problem in container terminals, and explain why this formulation cannot be used in context of bulk ports. The reason why we choose this particular formulation for comparison is that their problem definition comes closest to the problem we are addressing in this paper. Please note that the parameter names have been changed to match the ones that have already been defined in the previous section. The formulation is as follows:

$$\min \sum_{i\epsilon N} \sum_{k\epsilon M} (m_i - A_i + c_{ki}) x_{ki} \tag{4}$$

$$\text{s.t.} \sum_{k\epsilon M} x_{ki} = 1 \qquad \forall i \epsilon N \tag{5}$$

$$m_i - A_i \geq 0 \qquad \forall i \epsilon N \tag{6}$$

$$\sum_{k\epsilon M} (d_k - D_i) x_{ki} \geq 0 \qquad \forall i \epsilon N \tag{7}$$

$$\sum_{k\epsilon M} (\ell_k - \sum_{i,i'\epsilon N, i\neq i'} L_{i'} y_{ii'} x_{ki'} - L_i) x_{ki} \geq 0 \qquad \forall i \epsilon N \tag{8}$$

$$(m'_i + \sum_{k\epsilon M} c_{ki'} x_{ki'} - m_i)(m_i + \sum_{k\epsilon M} c_{ki} x_{ki} - m'_i) y_{ii'} x_{ki'} \geq 0 \quad \forall i \epsilon N, \forall k \epsilon M, i \neq i' \tag{9}$$

$$(m'_i + \sum_{k\epsilon M} c_{ki'} x_{ki'} - m_i)(m_i + \sum_{k\epsilon M} c_{ki} x_{ki} - m'_i)(1 - y_{ii'} x_{ki'}) \leq 0 \quad \forall i \epsilon N, \forall k \epsilon M, i \neq i' \tag{10}$$

$$x_{ki} \;\epsilon\; \{0, 1\} \qquad \forall i \epsilon N, \forall k \epsilon M \tag{11}$$

$$y_{ii'} \;\epsilon\; \{0, 1\} \qquad \forall i, i' \epsilon N \tag{12}$$

$$m_i \;\epsilon\; Z \qquad \forall i \epsilon N \tag{13}$$

In the above formulation, m_i is the starting time of the service of vessel i, c_{ki} is the handling time of vessel i at berth k, x_{ki} = 1 if vessel i is serviced at berth k, x_{ki} = 0 otherwise, $y_{ii'}$ = 1 if vessel i begins its service when vessel i' is being serviced at the same berth, $y_{ii'}$ = 0 otherwise. The input parameters are as defined earlier in the paper. The above formulation assumes that multiple vessels can be simultaneously serviced at the same berth, while each vessel occupies exactly one berth. In context of bulk ports, this understandably leads to less efficient space allocation of vessels. For example consider the case when the unoccupied length of a given berth having access to a specialized facility is smaller than the length of the vessel requiring this facility. In such a situation the entire berth is rendered useless for this vessel until the occupied portion of the berth is available again, thus increasing the waiting time for the vessel. The handling times are deterministic and fixed for a given berth and vessel, and thus the model also does not capture the variability in handling times with change in the berthing position of the vessel within the same occupied berth. Moreover, if only a certain part of a berth has access to a particular facility such as conveyor or pipeline, additional variables need to be introduced in the model to ensure proper allocation of vessels requiring these facilities. There are also several non-linearities in the model because of which the model cannot be easily solved using commercially available solvers for even small sized instances, and thus the authors have proposed a genetic algorithm based heuristic to solve the problem.

We now proceed with the description of the algorithms we have developed to solve the hybrid, dynamic berth allocation problem in context of bulk ports, and address the shortcomings of the above formulation.

4.1 MILP formulation

In this section, we present the MILP model for the dynamic, hybrid BAP in bulk ports. All temporal variables including the start time of operations and handling time of vessel, are modeled as continuous variables. The model uses several decision variables to obtain the berthing assignment of vessels to sections along the quay as well as the berthing order of vessels at each section, as shown below:

m_i ≥ 0, represents the starting time of handling of vessel $i \in N$;
c_i ≥ 0, represents the total handling time of vessel $i \in N$;
s_k^i binary, equals 1 if section $k \in M$ is the starting section of vessel $i \in N$, 0 otherwise;
x_{ik} binary, equals 1 if vessel $i \in N$ occupies section $k \in M$, 0 otherwise;
y_{ij} binary, equals 1 if vessel $i \in N$ is berthed to the left of vessel $j \in M$ without any overlapping in space, 0 otherwise;
z_{ij} binary, equals 1 if handling of vessel $i \in N$ finishes before the start of handling of vessel $j \in N$, 0 otherwise;

In the proposed formulation, we use additional parameters that are generated by data-preprocessing that provide information on whether a particular section is occupied and the fraction of the length of the section that is occupied for a given vessel and given the first section occupied by the vessel. The following coefficients are generated and provided as input to the model:

$$d_{ilk} = \begin{cases} 1 & \text{if vessel } i \text{ starting at section } \ell \text{ touches section } k; \\ 0 & \text{otherwise.} \end{cases}$$

$$p_{ilk} = \Big\{ \text{ percentage of total cargo handled at section } k \text{ if vessel } i \text{ starts at section } \ell;$$

The MILP model for the dynamic berth allocation problem with hybrid berth layout in bulk ports is formulated as shown below.

$$\min \sum_{i} (m_i - A_i + c_i) \tag{14}$$

$$\text{s.t. } m_i - A_i \geq 0 \qquad \forall i \epsilon N \tag{15}$$

$$\sum_{k \epsilon M} (s^j_k b_k) + B(1 - y_{ij}) \geq \sum_{k \epsilon M} (s^i_k b_k) + L_i \qquad \forall i, j \epsilon N, i \neq j \tag{16}$$

$$m_j + B(1 - z_{ij}) \geq m_i + c_i \qquad \forall i \epsilon N, \forall j \epsilon N, i \neq j \tag{17}$$

$$y_{ij} + y_{ji} + z_{ij} + z_{ji} \geq 1 \qquad \forall i \epsilon N, \forall j \epsilon N, i \neq j \tag{18}$$

$$\sum_{k \epsilon M} s^i_k = 1 \qquad \forall i \epsilon N \tag{19}$$

$$\sum_{k \epsilon M} (s^i_k b_k) + L_i \leq L \qquad \forall i \epsilon N \tag{20}$$

$$\sum_{\ell \epsilon M} (d_{ilk} s^i_\ell) = x_{ik} \qquad \forall i \epsilon N, \forall k \epsilon M \tag{21}$$

$$(d_k - D_i) x_{ik} \geq 0 \qquad \forall i \epsilon N, \forall k \epsilon M \tag{22}$$

$$c_i \geq h^w_k p_{ilk} Q_i s^i_\ell \qquad \forall i \epsilon N, \forall k \epsilon M, \forall l \epsilon M, \forall w \epsilon W_i \tag{23}$$

$$s^i_k \,\epsilon\, \{0, 1\} \qquad \forall i \epsilon N, \forall k \epsilon M \tag{24}$$

$$x_{ik} \,\epsilon\, \{0, 1\} \qquad \forall i \epsilon N, \forall k \epsilon M \tag{25}$$

$$y_{ij} \,\epsilon\, \{0, 1\} \qquad \forall i, j \epsilon N \tag{26}$$

$$z_{ij} \,\epsilon\, \{0, 1\} \qquad \forall i, j \epsilon N \tag{27}$$

The objective function (14) minimizes the total service time of all vessels berthing at the port. Constraint (15) ensures that vessels can be serviced only after their arrival. Constraints (16)-(18) are the non-overlapping restrictions for any two vessels berthing at the port. Note that the constraints (16)-(17) have been linearized by using a large positive constant B. Constraints 19)-(21) ensure that each vessel occupies only as many number of sections as determined by its length and the starting section occupied by the vessel. Note that using a hybrid berthing layout in case of bulk ports is important. On the one hand, a discrete berthing layout will lead to inefficient allocation of vessels since the delays associated with the vessels requiring fixed specialized equipment facilities such as conveyors and pipelines will further increase. On the other hand, a continuous berthing layout introduces several non-linearities in the modeling of the handling times of vessels which are dependent on the berthing location of the vessels along the quay. Moreover in the hybrid berthing layout used in our model, the sections are small enough such that a given vessel can occupy several sections while a given section can be occupied by at most one vessel at a given time. This maximizes the utilization of quay space ensuring better service quality of vessels requiring specialized equipment facilities such as conveyors and pipelines. Constraints (22) ensure that the draft of the vessel does not exceed the draft of any occupied section. Constraints (23) are used to determine the total handling time for any given vessel. The time taken to handle a given vessel at a given section is directly proportional to both the cargo quantity handled at that section as given by the product $p_{ilk}Q_i$, and the unit handling time associated with that section and cargo type given by the parameter h_k^w. The handling time at a section that is not occupied by the vessel is equal to zero as taken care of by the binary variable s_ℓ^i , and the total handling time of the vessel is the handling time at the section with the maximum handling time value among all sections along the quay. Note that the above model is completely linear and thus small instances can be solved using commercially available solvers.

The discrete BAP has been proved to be NP-hard (Garey and Johnson (1979)). The complexity of the proposed MILP based on hybrid berth layout is significantly enhanced because of the non-overlapping constraints (16)-(18), which contain several binary integer variables. This is because several combinations of specific integer values for the variables must be tested, and the number of such combinations rises exponentially with the size of the problem. Thus with increase in problem size, the space and time complexity of the MILP also increases exponentially.

4.2 GSPP formulation

The berth allocation problem described in Section (2) can also be modeled as a generalized set partitioning problem (GSPP). The GSPP model was proposed by Christensen and Holst (2008) in the context of container terminals. In the GSPP model, the planning horizon H is divided into discrete time intervals and only integral measurements of time are considered. The columns (variables) are generated apriori by data pre-processing, where a column represents a feasible berthing assignment of a single vessel to a specific set of section(s) at a specific time. In our research problem, the cargo type on the vessel is explicitly taken into account in the generation of feasible assignments for that particular vessel. For example, if a particular vessel carrying liquid bulk needs to be berthed at sections where the pipeline facility is installed, then only those assignments where the vessel is occupying these sections and the estimated departure time of the vessel does not exceed the length of planning horizon, are feasible. Alternatively, these vessels can be prevented from occupying sections that do not have the pipeline facility by providing extremely large handling time values associated with these sections and the cargo type corresponding to the conveyor facility. We illustrate the procedure with a small example containing two vessels 1 and 2. We consider 3 sections along the quay and 3 discrete time intervals in the planning horizon. Let us assume that vessel 1 needs the conveyor facility and cannot berth at section 3 which does not have this facility, while vessel 2 arrives at the start of time 2, and hence can only berth after that. Then the assignment matrix for the problem would look like as shown in Table 1. The first column represents the berthing assignment of vessel 1 to sections 1 and 2 from time 1-2, and

Vessel 1	1	1	0	0
Vessel 2	0	0	1	1
Section 1, Time 1	1	0	0	0
Section 1, Time 2	1	1	1	0
Section 1, Time 3	0	1	1	0
Section 2, Time 1	1	0	0	0
Section 2, Time 2	1	1	1	1
Section 2, Time 3	0	1	1	1
Section 3, Time 1	0	0	0	0
Section 3, Time 2	0	0	0	1
Section 3, Time 3	0	0	0	1

Table 1: Assignment matrix for a simple example of GSPP

so on.

We denote the set of columns by P. The assignment matrix is composed of the upper submatrix A and lower submatrix B. The upper submatrix A consists of $|P|$ columns and N rows. In submatrix A, if column $p \in P$ represents the feasible assignment of vessel $i \in N$, then the entry in row i is 1 while all other entries are zeroes. The lower submatrix B consists of $|P|$ columns and a single row for every (section,time) position. Thus, in submatrix B, if column $p \in B$, represents the feasible assignment of vessel $i \in N$, then all entries corresponding to the (section, time) positions occupied by vessel i in the feasible assignment $p \in P$ are 1, while all the remaining entries are zeroes.

We assume the following input data to be available for the GSPP model:

$H =$ set of discrete time intervals in the planning horizon
$P =$ set of feasible assignments
$t =$ $1, ..., |H|$ discrete time intervals in the planning horizon
$p =$ $1, ..., |P|$ feasible assignments
$d_p =$ delay associated with assignment p
$h_p =$ handling time associated with assignment p

The assignment matrix coefficients are defined as follows.

$$A_{ip} = \begin{cases} 1 & \text{if vessel } i \text{ is the assigned vessel in the feasible assignment represented} \\ & \text{by assignment } p; \\ 0 & \text{otherwise.} \end{cases}$$

$$b_p^{st} = \begin{cases} 1 & \text{if section } s \text{ is occupied at time } t \text{ in assignment } p; \\ 0 & \text{otherwise.} \end{cases}$$

There is only a single decision variable in the GSPP model for selection of feasible assignments in the optimal solution which is defined as follows.

$$\lambda_p = \begin{cases} 1 & \text{if assignment } p \text{ is part of the optimal solution;} \\ 0 & \text{otherwise.} \end{cases}$$

The GSPP model is formulated as shown below:

$$\min \sum_p (d_p\lambda_p + h_p\lambda_p) \tag{28}$$

$$\text{s.t.} \sum_p (A_{ip}\lambda_p) = 1 \qquad \forall i \epsilon N \tag{29}$$

$$\sum_p (b_p^{kt}\lambda_p) \leq 1 \quad \forall k \epsilon M, \forall t \epsilon H \tag{30}$$

$$\lambda_p \ \epsilon \ \{0,1\} \qquad \forall p \epsilon P \tag{31}$$

Constraints (29) ensure that each vessel must have exactly one feasible assignment in the optimal solution. Constraints (30) ensure that a given section at a given time can be occupied by at most one vessel

GSPP is in general characterized as NP-hard. The growth in the number of variables and constraints in the set-partitioning approach is much faster as compared to the integer programming approach. Furthermore constraints involving two or more vessels are much more difficult to incorporate in the GSPP approach. For example in bulk context, it would be complicated to model constraints wherein two or more vessels cannot use the pipeline facilties installed at different sections at the same time. However, the approach offers several modeling advantages, primarily because it is much easier to incorporate more advanced spatial and temporal constraints on individual vessels as these can be easily handled while generating feasible assignments. It is also easier to model complex objectives as long as they can be expressed as a function of the column costs.

4.3 Heuristic Algorithm

We now propose a meta-heuristic that improves the berthing assignment of vessels by iteratively changing the priority order of vessels with regards to the service quality measure of each vessel. The algorithm, commonly termed in literature as squeaky wheel optimization (SWO) works on the principle of Construct/ Analyze/ Prioritize, where the solution generated at each successive iteration is constructed and analyzed, and the results of this analysis are used to generate a new priority order to obtain the next solution. The algorithm operates on two search spaces: solutions and priorizations as schematically shown in Figure 5. The idea of SWO was introduced by Clements et al. (1997) and has been used in several combinatorial optimization problems such as in scheduling problems and graph coloring problems, by Smith and Pyle (2004), Lim et al. (2004), Joslin and Clements (1998). In the context of container terminals, SWO algorithm has been used by few scholars such as Fu et al. (2007) to solve the port space allocation problem, and by Meisel and Bierwirth (2008) to solve the integrated berth allocation and crane assignment problem. The approach can be adapted to solve the dynamic, hybrid berth allocation problem in bulk ports as discussed in this section. SWO is typically useful in problems where it is possible to quantify the individual contribution of each single problem element to the overall solution quality. It could be used in our problem, since the objective is to minimize the total service time of the berthing schedule which is simply the sum of the service times of all vessels in the berthing schedule. Unlike in local search techniques such as iterated hillclimbing etc., the moves in search space are not motivated by the objective function value, but rather by the weak performing elements of the solution even when the move may lead to a worse overall solution.

In the implementation of the SWO algorithm in our problem, we use a base heuristic that returns a feasible berthing assignment for a given priority order of vessels. In the generation of feasible assignments, the cargo type on the vessel is explicitly taken into account. The feasible assignments of vessels requiring fixed equipment facilities such as conveyors and pipelines have the vessels berthed at only those sections where these facilities are installed, or alternatively we can provide extremely large values of handling times for other sections along the quay. The initial solution is the berthing assignment obtained by prioritizing the incoming vessels in order of arrival times, also called the first-come-first-served (FCFS) ordering. At the end of each iteration, we assess the individual contribution of each vessel to the overall service time of the berthing assignment obtained, and rank the vessels according to their individual service performance to obtain a new priority order which is then used as an input in the next iteration. A vessel may be inserted between two vessels that have already been assigned or at the beginning or at the end of the berthing schedule. A given vessel at a particular rank in the given priority order is assigned to those set of sections which minimize the total waiting and handling time of the vessel, after all the vessels ranked above in the priority order have already been assigned.

Algorithm 1 Algorithm for implementation of SWO to solve BAP

```
Require: Set $N$ of vessels, set $M$ of sections
  Obtain the initial priority list $p_o$ given by increasing order of arrival times
  Initialize arraylist priorityListsEvaluated
  Add $p_o$ to priorityListsEvaluated
  Set $p_o \rightarrow$ current priority list $p$
  while iteration $\leq$ maxiterations do
    Obtain the berthing assignment $b$ = BaseHeuristic($p$)
    for $i = 1 \rightarrow N$ do
      Calculate service quality $s_i$ of vessel $i$ in solution $b$
    end for
    Sort the vessels in decreasing order of $s_i$ to obtain the new priority list $p_1$
    Set $p_1 \rightarrow p$
    while priorityListsEvaluated contains $p$ do
      $p$ = Randomize ($p$)
    end while
    Add $p$ to priorityListsEvaluated
    if service time($b$) $\leq$ service time(finalBerthingAssignment) then
      finalBerthingAssignment = $b$
    end if
  end while
```

The key role of the prioritizer is to identify the weak performing vessels and move them forward in the sequence to enable them to be handled better by the constructor. Once these vessels start performing well, they sink back in the sequence at which point their performance may start deteriorating again and cause them to move forward in the sequence again. On the other hand, vessels performing consistently well sink back and stay there. In the bulk context, owing to the restrictive spatial constraints on vessels requiring specialized facilities such as conveyors and pipelines, the berthing delays for these vessels could be very large and these vessels are typically the weakest performing elements in the overall solution. Such vessels are heavily penalized if they are ranked lower in the priority list, and thus to ensure high service quality for these vessels in particular, they should not be allowed to sink back in the priority list. In the SWO algorithm, there is also the risk of the algorithm getting trapped in a cycle alternating between a set of priority listings. To avoid this, if a listing has already been evaluated, we generate a new listing by introducing some randomization in the current priority order by swapping two or more vessels until we get a priority listing that has not been evaluated so far. The algorithm terminates after a preset number of iterations, and the best solution obtained thus far is accepted as the final solution. The implementation is described by Algorithm 1.

One key issue in using SWO for large problem size is that a new solution is constructed from scratch after every iteration. A possible solution to this problem could be the use of a history mechanism that keeps track of the previous solutions generated for given prioritizations, as that would speed up the construction process. Another bottleneck in the approach is that in many cases, the optimal solution has some badly performing vessels. This is a problem since this approach is primarily motivated by identifying such vessels and assigning high blame to them to move them forward in the sequence and enable them to be handled better in the next iteration. This prevents the SWO approach to identify such sacrificial vessels and converge to good solutions. While a deeper understanding of the approach would definitely help to obtain better solutions, in this paper we have shown that a relatively simple implementation of the approach can be used to obtain reasonably good results for our problem.

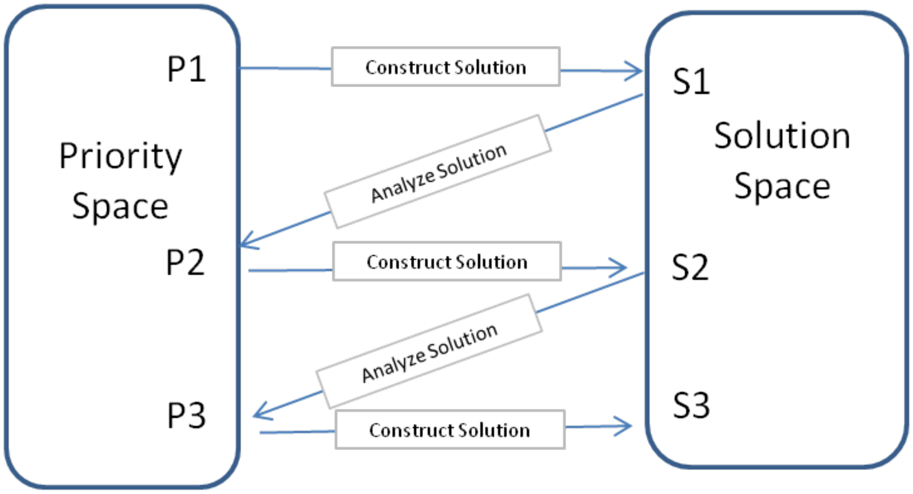

Figure 5: *Schematic Representation of SWO Algorithm*

5 Results and Analysis

In this section, we compare the different BAP formulations presented in earlier sections. The MILP formulation described in Section 4.1 is tested using CPLEX solver with the solution time limit set to 7200 seconds. In the GSPP approach, all feasible assignments for the given instances are generated apriori using JAVA code and provided as an input to the GSPP formulation described in Section 4.2. The optimization model in (28)- (31) is then solved using CPLEX solver. The heuristic algorithm presented in Section 4.3 is implemented in JAVA programming language. All tests were run on an Intel Core i7 (2.80 GHz) processor and used a

32-bit version of CPLEX 12.2.

5.1 Generation of Instances

The instances were generated based on a small sample of data received from SAQR port, Ras Al Khaimah(RAK), UAE. SAQR port is the biggest bulk commodity port in the entire middle east handling approximately 30 million tonnes of cargo annually. The port plays a key role in the economic growth of the RAK emirate, which has registered a significant growth in GDP from AED 6.6 billion in 2002 to AED 13.6 billion in 2008. The port's cargo handling department specializes in dealing with a wide variety of imported and exported commodities including consignments of aggregates, cement, coal, clinker, iron ore, feldspar, clay, soda ash, silica sand, grain, animal feedstock, steel, project cargoes and petroleum products (www.saqrport.com).

The data sample received from the port provided information about the vessel lengths, expected and actual times of arrival, berthing, processing and departure of vessels, expected and actual berthing positions and the cargo tonnage of the vessels. The data was provided for over 20 vessels for a time horizon of roughly 10 days from 28th March to 6th April, 2011. Although, there were a lot of missing entries in the data file, we could use the data sample to get a rough estimate of the range of values for most input parameters in our model.

To do a comparison of the different formulations, we generate 6 instance sizes with $|N| =$ 10, 25 and 40 vessels and $|M| = 10$ and 30 sections along the quay. A set of 9 instances was generated for each instance size. In all instances, the total quay length L is 1600 meters, and the vessel lengths L_i lie in the range 80-260 meters as in SAQR. The expected arrival times A_i are randomly generated between a given range of values, described more explicitly later in the paper. We further remark that the drafts of all vessels D_i are less than the minimum draft for all sections, as in the data provided by the port. Therefore, constraints (22) are never active for the tested instances.

In the generation of handling times h_k^w, we consider six cargo types - clay, grain, coal, cement, conveyor and pipeline. The crane handling rate T is assumed equal to 1000 tonnes per hour. The number of cranes operating in section k is determined by the length of the section, assuming an additional crane for every 50 meters of section length. In Figure 6, the cargo locations on the yard with respect to the quay axis have been graphically shown. The distance d_k^w in equation 3 is calculated as the euclidean distance between the midpoint of section k and the cargo location w. The parameter v^w which is dependent on the rate of transfer of cargo is assumed to be equal to $1/1200$ hours per meter per unit cargo for the conveyor, $1/3600$ hours per meter per unit cargo for the pipeline and $1/600$ hours per meter per unit cargo for all other cargo types.

The modeling results were found to be sensitive to the discretization used in the berthing

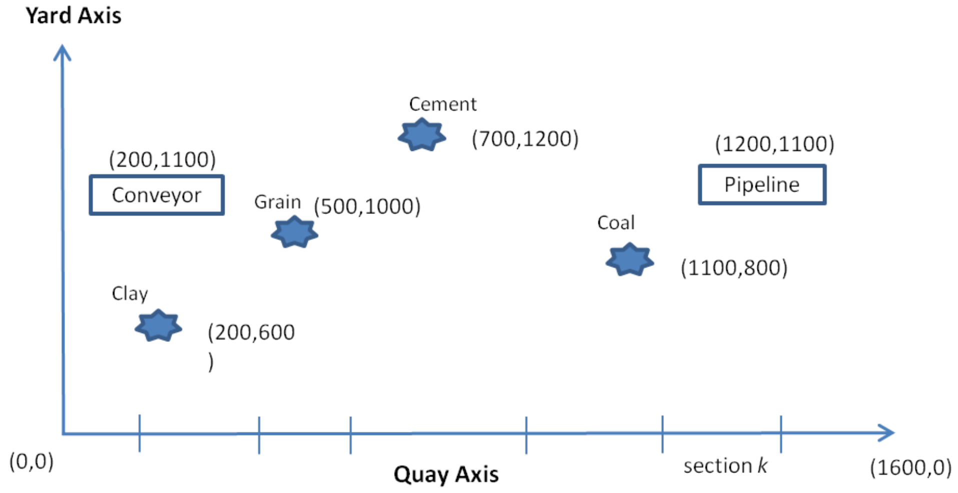

Figure 6: *Graphical Representation of the yard layout used in the generation of handling times*

layout, even for the same number of sections. In the instances shown, the berthing layouts have been fixed for |M| = 10, 30. The minimum section length is 25 meters, and all section lengths are whole number multiples of this minimum section length. We use the berthing layouts as shown in Table 2 and Table 3.

To test the sensitivity of the results with respect to the different input parameters in the model, the instances have been desgined in the following manner.

1. Within the same instance id, the instances have been indexed from 1 to 9. Instances with the same index number in instance ids A and B, have all parameters the same except the berthing layouts which consist of |M| = 10 sections in instance id A and |M| = 30 sections in instance id B. Similarly for C and D, E and F.

2. Within the same instance id, say A, instances with indices (1, 2, 3), (4, 5, 6) and (7, 8, 9) have all parameters the same except the vessel lengths, which vary in the range 80-260 meters for instances indexed (1,4,7); in the range 80-170 meters for indices (2,5,8) and in the range 170-260 meters for indices (3,6,9). Similarly for B, C, D, E and F.

3. Within the same instance id, say A, instances with indices (1, 4, 7), (2, 5, 8) and (3, 6, 9) have all parameters as same except the arrival times. Indices (1,2,3) represent the congested scenario wherein the vessel arrivals are very close together, within a time range of 5 hours; indices (4,5,6) represent the mildly congested scenario with vessel arrival times within a time range of 10 hours; and indices (7,8,9) represent the congestion free case when the vessel arrivals are spread within a time range of 20 hours. Similarly for B, C, D, E and F.

Section	Length	Facility
1	150	C
2	50	C
3	200	P
4	150	C,P
5	125	C,P
6	250	-
7	250	-
8	75	P
9	150	P
10	200	-

Table 2: Berthing layout and fixed facility locations for |M|=10 (C and P stand for conveyor and pipeline respectively)

Thus instance A8 has a berthing layout with |M|=10 sections, |N|=10 vessels with vessel lengths in the range 80-170 meters and represents the congestion free scenario with all the vessels arriving within a time range of 20 hours. Instance D6 contains |M|=30 sections, |N|=25 vessels with vessel lengths in the range 170-260 meters and represents the mildly congested scenario with vessel arrivals within a time range of 10 hours. Instance F1 contains |M|=30 sections, |N|=40 vessels with vessel lengths in the range 80-260 meters and represents the congested scenario with all vessel arrivals within a time window of 5 hours.

5.2 Comparison of Algorithms

The computational results for the three approaches presented earlier are shown in Tables (4)-(5). Results obtained from the first-come-first-served (FCFS) heuristic which is used to obtain an initial solution in the implementation of the SWO heuristic algorithm have been also shown.

As can be seen from the results, for |N| = 10 vessels, the MILP formulation produces optimal results for all instances within the prescribed CPLEX time limit of 2 hours. However, it is not able to solve even a single instance to optimality for instances with |N| = 25, 40, with a significantly large duality gap at the end of the run. For instance F3 which represents the congested scenario with |N| = 40 vessels and |M| = 30 sections, the model is unable to find even a single feasible integer solution within the CPLEX time limit. Clearly, the complexity of the problem is highly affected by the problem size and increases exponentially, which as discussed earlier can be attributed to the exponentially increasing number of integer variables with increase in problem size.

In the GSPP formulation, we generate feasible assignments for a sufficiently large plan-

Section	Length	Facility
1	150	C
2	25	C
3	25	C
4	25	P
5	25	P
6	100	P
7	25	P
8	25	P
9	25	C,P
10	25	C,P
11	100	C,P
12	25	C,P
13	25	C,P
14	25	C,P
15	50	C,P
16	50	-
17	75	-
18	200	-
19	75	-
20	50	-
21	50	-
22	25	P
23	25	P
24	25	P
25	100	P
26	25	P
27	25	P
28	25	-
29	25	-
30	150	-

Table 3: Berthing layout and fixed facility locations for |M|=30 (C and P stand for conveyor and pipeline respectively)

ning horizon of 150 hours, divided into discrete time intervals of 1 hour. It should be noted that the computational time provided for GSPP model includes the time taken to generate the feasible assignments and subsequently solve the optimization model using CPLEX. As can be seen from the results, the performance of the GSPP model is quite remarkable, as it is able to solve all instances to optimality, and most of them within few minutes of computational time. For instance id F with $|N| = 40$ and $|M| = 30$, the GSPP model runs out of memory when the length of time interval h is equal to 1 hour, since the number of feasible assignments is very large. To overcome this problem, we use time intervals of h=2 hours for instance id F.

The FCFS heuristic produces results by simply berthing vessels according to their arrival order. The heuristic used to obtain an initial solution in the implementation of the SWO, performs reasonably well for small sized instances but the performance is weak for larger instances. This indicates that as instance size grows, the berthing order has a larger deviation from the arrival order of vessels.

The SWO heuristic performs reasonably well for the tested instances. The optimality gap is less than 10 percent averaged over all the tested 54 instances. Since the feasible assignments are explicitly enumerated in the GSPP approach, it runs the risk of running out of memory for a large number of assignments. Even when the time interval h is equal to 2 hours, the GSPP formulation takes a relatively larger time to converge for some instances, as can be specially seen for instances $F3$, $F6$ and $F9$ which represent the instances with large vessels with $|N| = 40$ vessels and $|M| = 30$ sections. But the SWO heuristic provides near optimal solutions in much less time as compared to GSPP. Thus, for very large instances or even smaller sized instances with long planning horizons, SWO may be used to obtain sub-optimal berthing assignments.

5.3 Results Analysis

In Figures (7)-(8), the optimal solutions of the instances $A1$ and $C1$ are graphically represented on the time space diagram as shown. Each rectange represents a vessel whose width represents the berthing location of the vessel along the quay, and the height represents the handling time of the vessel. The cargo type on each vessel is marked on the rectangle representing that particular vessel. The fixed facility locations are marked along the quay axis. It can be seen that the optimal berthing locations for most vessels are at close physical proximity to the location of the vessel cargo type on the yard, as per the yard layout shown in Figure 6. This is the case since the distance between the berthing location of the vessel and the yard location of the cargo type of the vessel is explicitly considered in modeling the handling times of the vessels. The vessels requiring specialized equipment facilities are berthed at sections where these facilities are installed by providing appropriate values of handling

Table 4: Computational results for generated instances, Instances A, B, C

Instance	MILP			GSPP		FCFS		SWO		
	obj	gap[a]	time[b]	obj	time(|H|=150, t=1)[c]	obj	RE[d]	obj	RE[d]	time
10x10										
A1	230.21	0.01%	67.67	231.21	5.94	262.09	13.36%	230.48	-0.32%	15.81
A2	237.35	0.01%	15.31	238.49	5.54	250.44	5.01%	239.08	0.25%	16.66
A3	223.99	0.01%	9.58	226.61	5.96	280.04	23.58%	225.33	-0.57%	16.97
A4	227.12	0.01%	10.31	227.22	5.68	240.91	6.03%	228.00	0.35%	16.60
A5	234.20	0.01%	5.60	234.22	5.43	251.09	7.20%	234.47	0.11%	16.30
A6	233.12	0.01%	11.06	234.06	6.85	262.61	12.20%	233.12	-0.40%	16.90
A7	203.23	0.00%	0.56	203.23	4.99	208.44	2.57%	203.38	0.07%	15.95
A8	218.87	0.00%	0.56	219.99	5.29	220.90	0.41%	218.87	-0.51%	16.72
A9	198.03	0.00%	0.60	199.89	5.16	214.17	7.14%	198.03	-0.93%	17.53
Mean							8.61%		-0.22%	
10x30										
B1	188.39	0.01%	15.80	189.73	94.552	216.56	14.14%	192.81	1.62%	49.95
B2	178.07	0.01%	15.78	179.10	86.08	186.41	4.08%	178.08	-0.57%	48.42
B3	200.16	0.01%	1094.61	202.33	101.93	230.14	13.74%	216.14	6.82%	49.79
B4	182.57	0.01%	3.04	184.27	89.58	224.00	21.56%	182.80	-0.80%	47.73
B5	178.48	0.01%	10.97	179.23	85.01	185.60	3.55%	179.01	-0.12%	48.37
B6	199.82	0.01%	87.78	201.17	96.19	240.09	19.35%	223.37	11.04%	48.83
B7	173.02	0.01%	1.30	173.02	86.00	175.72	1.56%	175.30	1.32%	48.61
B8	162.51	0.00%	1.57	162.81	81.67	169.20	3.92%	166.20	2.08%	50.29
B9	175.29	0.00%	1.39	175.81	95.74	192.41	9.44%	191.26	8.79%	50.27
Mean							10.15%		3.35%	
25x10										
C1	812.32	33.08%	-	819.22	14.09	976.49	19.20%	869.31	6.11%	22.29
C2	783.45	30.27%	-	781.72	11.70	924.35	18.25%	825.92	5.65%	22.58
C3	903.51	33.39%	-	900.43	20.19	1107.59	23.01%	929.32	3.21%	22.24
C4	795.71	23.18%	-	791.18	15.26	877.90	10.96%	852.03	7.69%	23.28
C5	751.19	27.47%	-	747.88	10.41	846.26	13.15%	774.17	3.51%	22.16
C6	874.53	24.50%	-	863.86	19.38	979.26	13.36%	898.44	4.00%	23.19
C7	735.13	19.72%	-	741.16	15.91	840.85	13.45%	806.23	8.78%	22.54
C8	689.37	22.26%	-	699.14	11.23	761.48	8.92%	735.46	5.20%	22.32
C9	800.00	22.76%	-	793.24	12.82	936.55	18.07%	872.76	10.03%	23.56
Mean							15.37%		6.02%	

[a] The gap is calculated with respect to the value of linear relaxation as (Upper bound - LP lower bound)/ Upper bound.

[b] "-" indicates that the CPLEX time limit of 2 hours was reached. "+" indicates that CPLEX was unable to find a feasible integer solution within the time limit.

[c] "*" indicates that the CPLEX solver ran out of memory for t=1 hour, and results were calculated for t = 2 hours to reduce the number of feasible assignments

[d] The relative error, RE is calculated with respect to the objective function value of GSPP model

Table 5: Computational results for generated instances, Instances D, E, F

Instance	MILP			GSPP		FCFS		SWO		
	obj	gap[a]	time[b]	obj	time(|H|=150, t=1)[c]	obj	RE[d]	obj	RE[d]	time
25x30										
D1	690.79	23.14%	-	670.42	219.04	857.99	27.98%	785.91	17.23%	105.36
D2	617.31	34.23%	-	591.06	185.44	723.13	22.34%	667.20	12.88%	96.03
D3	809.55	30.75%	-	784.94	387.40	984.28	25.40%	866.82	10.43%	105.66
D4	657.48	20.62%	-	636.19	220.40	778.50	22.37%	728.60	14.53%	100.95
D5	560.65	25.96%	-	556.37	172.09	622.14	11.82%	614.72	10.49%	91.35
D6	754.87	21.97%	-	739.44	253.67	909.53	23.00%	836.23	13.09%	102.34
D7	581.54	8.36%	-	590.24	194.80	731.55	23.94%	706.82	19.75%	100.82
D8	510.80	16.20%	-	506.30	167.09	565.87	11.77%	565.87	11.77%	111.46
D9	704.76	19.18%	-	677.97	200.63	848.97	25.22%	778.67	14.85%	99.87
Mean							21.54%		13.89%	
40x10										
E1	1243.64	63.77%	-	1140.60	41.73	1536.78	34.73%	1289.88	13.09%	28.24
E2	1193.05	59.69%	-	1138.16	18.80	1571.07	38.04%	1273.09	11.86%	28.44
E3	1341.65	67.35%	-	1249.06	139.47	1878.78	50.42%	1416.54	13.41%	31.50
E4	1113.36	59.53%	-	1051.50	30.87	1408.95	33.99%	1137.20	8.15%	30.11
E5	1105.34	56.98%	-	1063.85	19.06	1447.39	36.05%	1202.50	13.03%	32.06
E6	1361.62	68.15%	-	1160.05	167.58	1903.39	64.08%	1330.64	14.71%	34.01
E7	1011.20	55.47%	-	946.35	26.04	1291.11	36.43%	1148.14	21.32%	31.69
E8	1013.41	53.02%	-	953.24	20.03	1183.57	24.16%	1094.15	14.78%	32.01
E9	1181.97	64.95%	-	1071.46	94.88	1500.71	40.06%	1296.79	21.03%	35.64
Mean							39.77%		14.60%	
40x30										
F1	1193.42	70.56%	-	920.73	506.02*	1278.04	38.81%	1092.44	18.65%	169.53
F2	913.59	62.66%	-	863.43	127.91*	1168.60	35.34%	911.47	5.56%	159.28
F3	+	+	+	1089.48	932.56*	1784.98	63.84%	1411.24	29.53%	173.89
F4	902.74	59.15%	-	856.41	3341.62*	1143.59	33.53%	1035.72	20.94%	160.113
F5	881.37	61.20%	-	786.27	137.91*	973.94	23.87%	857.47	9.06%	163.41
F6	1121.14	66.39%	-	1015.53	2281.78*	1628.76	60.39%	1286.73	26.71%	265.29
F7	922.04	62.05%	-	777.06	829.88*	932.34	19.98%	932.34	19.98%	166.52
F8	728.48	52.93%	-	679.58	131.81*	774.12	13.91%	745.00	9.63%	160.49
F9	934.35	58.59%	-	920.29	1767.57*	1458.45	58.48%	1214.66	31.99%	171.97
Mean							38.68%		19.12%	

[a] The gap is calculated with respect to the value of linear relaxation as (Upper bound - LP lower bound)/ Upper bound.

[b] "-" indicates that the CPLEX time limit of 2 hours was reached."+" indicates that CPLEX was unable to find a feasible integer solution within the time limit.

[c] "*" indicates that the CPLEX solver ran out of memory for t=1 hour, and results were calculated for t = 2 hours to reduce the number of feasible assignments

[d] The relative error, RE is calculated with respect to the objective function value of GSPP model

times for each combination of section and cargo type. Thus for a given yard layout of the bulk terminal our model ensures better coordination between the berthing and yard activities, apart from minimizing the total service time of all vessels berthing at the port. It may be noted however that some vessels requiring the fixed equipment facilities experience large delays owing to the restrictive spatial constraints on these vessels.

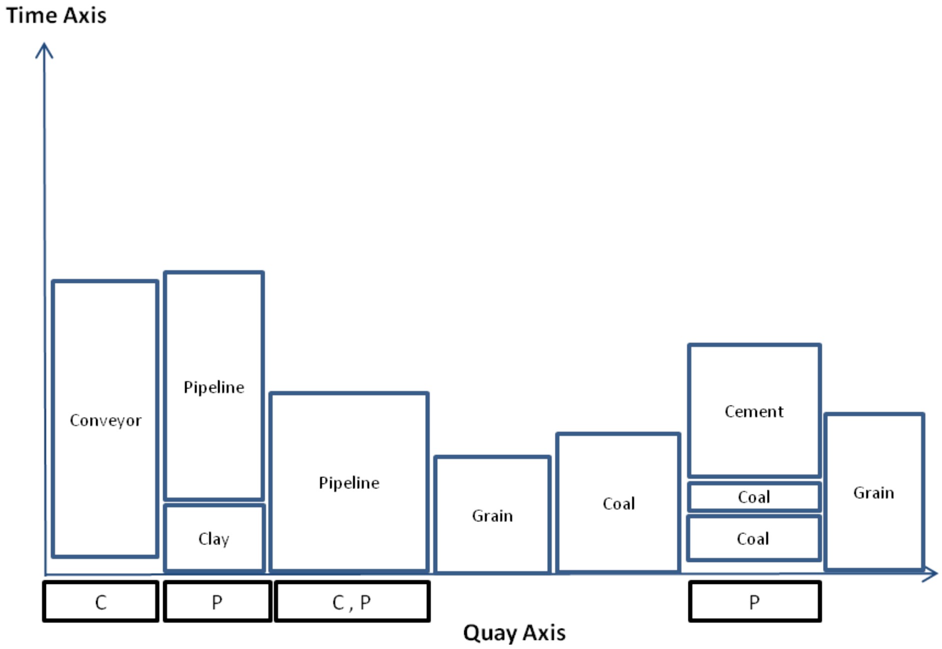

Figure 7: *Graphical Representation of the optimal solution output for the instance* A1*. The fixed facilitly locations are marked along the quay, where* C *and* P *stand for conveyor and pipeline respectively.*

The computational results obtained from the GSPP formulation are used to investigate the impact of vessel lengths, berthing layouts and congestion on the optimal service times and complexity of the BAP. In Figure (9), the percentage difference in optimal service time values for |M|=10 and |M|=30 are plotted for different number of vessels. It is clear from the plots that having more sections leads to better service times owing to better utilization of the quay space. It is interesting to note however that the difference in optimal service times is most significant for instances with smaller vessels represented by indices (2,5,8). This clearly indicates that instances with small vessels are more sensitive to the berthing layouts and choosing a higher number of sections is more advantageous for such instances. As can be seen in Table (5), in many cases for |N|=40 vessels, the computational time is significantly higher for |M|=30 sections as compared to |M|=10 sections. Thus, in choosing the discretization for larger problem size, there may be a trade-off between obtaining a better

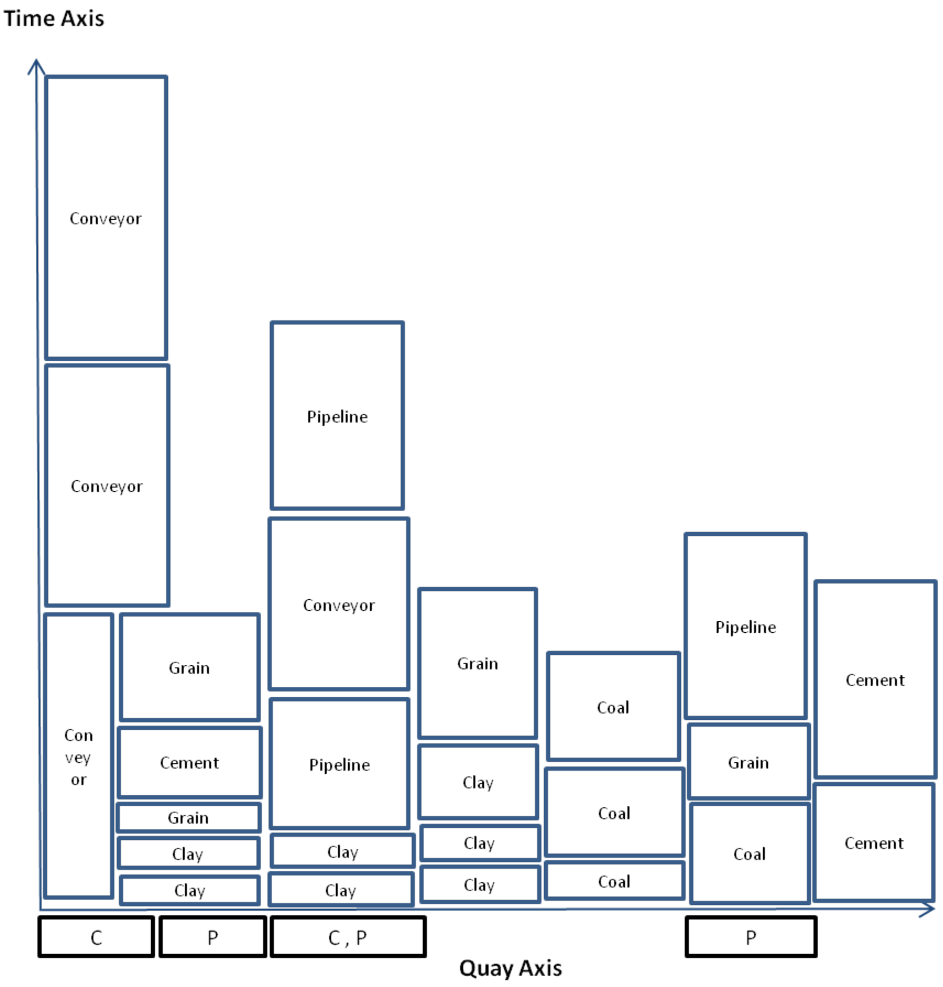

Figure 8: *Graphical Representation of the optimal solution output for the instance* C1. *The fixed facilitly locations are marked along the quay, where* C *and* P *stand for conveyor and pipeline respectively.*

solution value and obtaining the solution in reasonable computational time.

In Figure 10, the optimal service times have been plotted for each instance size with varying degrees of congestion for vessel lengths in the range 80-260 meters represented by instance ids (1,4,7). Here, congestion is used to represent the case when vessel arrivals are very close together in time, as opposed to the congestion free case when arrivals are widely spaced in time. It can be seen from the plot, that as instance size grows the effect of congestion is also higher as indicated by the negative slopes of the curves for the larger instances. Thus it can be inferred that the temporal proximity of vessel arrivals enhances the complexity of the problem, and leads to higher service time values.

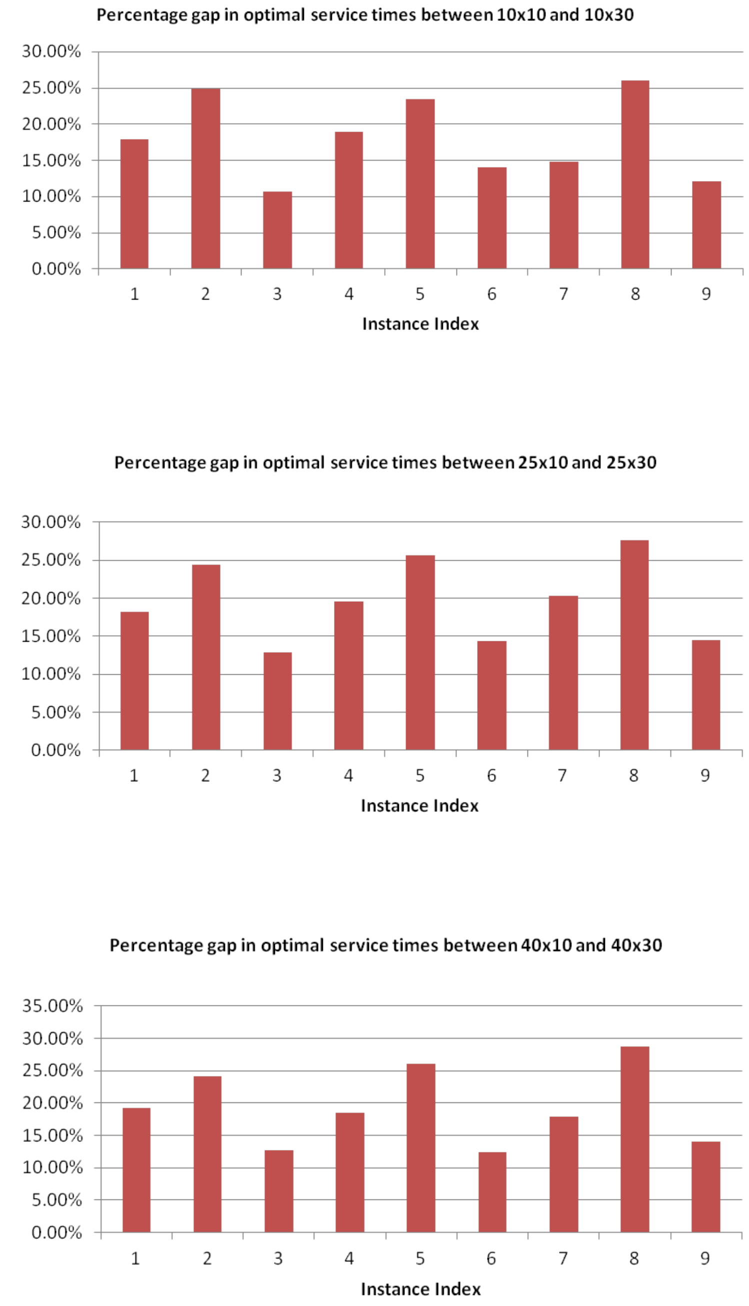

Figure 9: *Effect of discretization and vessel lengths on service times*

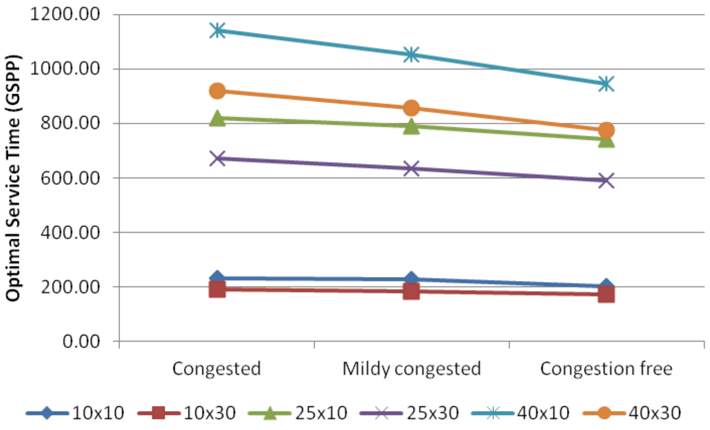

Figure 10: *Effect of congestion on service times*

6 Conclusions and Future Research

The berth allocation problem in bulk ports has many similarities but also few differences with that in container terminals. The main point of difference is that in bulk port terminals, it is necessary to explicitly account for the cargo type on the vessel and the fixed equipment facilities such as conveyors and pipelines which are installed at only certain sections along the quay. In this paper, we have presented and compared three different formulations to solve the dynamic, hybrid BAP in bulk port terminals that explicitly take into account the cargo type on the vessel. Our approach enhances the co-ordination between berthing and yard activities for a given yard layout and fixed facility locations along the quay, apart from minimizing the total service cost of all berthing vessels. Results inspired by port data show that the problem is complex. The exact solution approach based on mixed integer linear programming (MILP) fails to produce optimal results for large sized instances in the CPLEX time limit, though it can be used to solve small sized instances in reasonable time.

We have proposed an alternate exact solution approach based on generalized set partitioning (GSPP) in which all feasible assignments for a given planning horizon are generated a priori by data pre-processing and provided as an input to the optimization model. The performance of the GSPP model is quite remarkable. The model was able to solve all tested instances to optimality. However, the increase in the number of variables and constraints in the GSPP model is very fast with increase in the instance size. When the number of feasible assignments is too large, typically in excess of 100,000, which could be owing to large instance size or a relatively long planning horizon for even small instances, the CPLEX solver runs out of memory. This could be partially handled by tweaking the parameters of the C-PLEX solver, but was avoided for a fair comparison of the set partitioning method with the other approaches.

To obtain near-optimal solutions of large sized instances in reasonable time, we have proposed a heuristic approach based on the principle of squeaky wheel optimization. The heuristic produces sub-optimal results with less than 10 percent gap averaged over all the tested instances with respect to the GSPP solution. Thus, SWO algorithm could be used as an alternative in cases where GSPP model is too slow or does not provide any result.

Another challenging problem in the bulk context is to make the proposed model more robust to account for unforeseen disruptions in operations owing to uncertainties in arrival times and/or the breakdown of certain facilities such as conveyors or pipelines. As part of ongoing work, the berth allocation problem is modeled with some degree of anticipation of variability in information. The robust model is compared with the deterministic model presented in this paper by using a dynamic recovery algorithm that solves the berth allocation problem in real time as actual data is revealed. Another natural extension of the current work is to extend our berth allocation model and take into account the integrated planning of berth and yard space allocation.

References

Bierwirth, C. and Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals, *European Journal of Operational Research* **202**(3): 615–627.

Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J. and Lusby, R. (2011). Models for the discrete berth allocation problem: A computational comparison, *Transportation Research Part E* **47**(4): 461 – 473.

Chang, D., Yan, W., Chen, C.-H. and Jiang, Z. (2008). A berth allocation strategy using heuristics algorithm and simulation optimisation, *Int. J. Comput. Appl. Technol.* **32**: 272–281.

Cheong, C., Tan, K., Liu, D. and Lin, C. (2010). Multi-objective and prioritized berth allocation in container ports, *Annals of Operations Research* **180**: 63–103.

Christensen, C. G. and Holst, C. T. (2008). *Berth allocation in container terminals*, Master's thesis, Technical University of Denmark.

Clements, D., Crawford, J., Joslin, D., Nemhauser, G. L., Puttlitz, M. and Savelsbergh, M. W. P. (1997). Heuristic optimization: A hybrid ai/or approach, *Proceedings of the Workshop on Industrial Constraint-Directed Scheduling.*

Cordeau, J. F., Laporte, G., Legato, P. and Moccia, L. (2005). Models and tabu search heuristics for the berth-allocation problem, *Transportation Science* **39**(4): 526–538.

Dai, J., Lin, W., Moorthy, R. and Teo, C. P. (2008). Berth allocation planning optimization in container terminals, *in* C. S. Tang, C. P. Teo and K. K. Wei (eds), *Supply Chain Analysis*, Vol. 119 of *International Series in Operations Research & Management Science*, Springer, pp. 69–104.

Fu, Z., Li, C. L., Lim, A. and Rodrigues, B. (2007). Port space allocation with a time dimension, *JORS* **58**: 797807.

Gao, C., Zhang, R., Du, Y. and Chen, Q. (2010). A proactive and reactive framework for berth allocation with uncertainties, *Advanced Management Science (ICAMS), 2010 IEEE International Conference on*, Vol. 3, pp. 144 –149.

Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*, Freeman & co., New York.

Guan, Y. and Cheung, R. K. (2004). The berth allocation problem: models and solution methods, *OR Spectrum* **26**: 75–92.

Guan, Y., Xiao, W.-Q., Cheung, R. K. and Li, C.-L. (2002). A multiprocessor task scheduling model for berth allocation: heuristic and worst-case analysis, *Operations Research Letters* **30**(5): 343 – 350.

Han, X.-l., Lu, Z.-q. and Xi, L.-f. (2010). A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time, *European Journal of Operational Research* **207**(3): 1327 – 1340.

Imai, A., Nagaiwa, K. and Chan, W. T. (1997). Efficient planning of berth allocation for container terminals in Asia, *Journal of Advanced Transportation* **31**(1): 75–94.

Imai, A., Nishimura, E., Hattori, M. and Papadimitriou, S. (2007). Berth allocation at indented berths for mega-containerships, *European Journal of Operational Research* **179**(2): 579–593.

Imai, A., Nishimura, E. and Papadimitriou, S. (2001). The dynamic berth allocation problem for a container port, *Transportation Research Part B* **35**(4): 401–417.

Imai, A., Nishimura, E. and Papadimitriou, S. (2003). Berth allocation with service priority, *Transportation Research Part B* **37**(5): 437–457.

Imai, A., Nishimura, E. and Papadimitriou, S. (2008). Berthing ships at a multi-user container terminal with a limited quay capacity, *Transportation Research Part E* **44**: 136–151.

Imai, A., Sun, X., Nishimura, E. and Papadimitriou, S. (2005). Berth allocation in a container port: using a continuous location space approach, *Transportation Research Part B* **39**(3): 199–221.

Joslin, D. and Clements, D. (1998). "squeaky wheel" optimization, *AAAI'98 Proceedings of the national conference on Artifical intelligence*.

Kim, K. H. and Moon, K. C. (2003). Berth scheduling by simulated annealing, *Transportation Research Part B* **37**(6): 541–560.

Li, C. L., Cai, X. and Lee, C. Y. (1998). Scheduling with multiple-job-on-one-processor pattern, *IIE Transactions* **30**: 433–445.

Lim, A. (1998). The berth planning problem, *Operations Research Letters* **22**(2-3): 105–110.

Lim, A., Rodrigues, B., Xiao, F. and Zhu, Y. (2004). Crane scheduling with spatial constraints, *Naval Research Logistics* **51**: 386–406.

Mauri, G., Oliveira, A. and Lorena, L. (2008). A hybrid column generation approach for the berth allocation problem, *in* J. van Hemert and C. Cotta (eds), *Evolutionary Computation in Combinatorial Optimization*, Vol. 4972 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 110–122.

Meisel, F. and Bierwirth, C. (2008). Heuristics for the integration of crane productivity in the berth allocation problem, *Transportation Research Part E* **45**(1): 196–209.

Monaco, M. F. and Sammarra, M. (2007). The berth allocation problem: a strong formulation solved by a lagrangean approach, *Transportation Science* **41**(2): 265–280.

Moorthy, R. and Teo, C. P. (2006). Berth management in container terminal: the template design problem, *OR Spectrum* **28**(4): 495–518.

Nishimura, E., Imai, A. and Papadimitriou, S. (2001). Berth allocation planning in the public berth system by genetic algorithms, *European Journal of Operational Research* **131**(2): 282 – 292.

Park, K. T. and Kim, K. H. (2002). Berth scheduling for container terminals by using a sub-gradient optimization technique, *Journal of the Operational Research Society* **53**: 1054–1062.

Park, Y. M. and Kim, K. H. (2003). A scheduling method for berth and quay cranes, *OR Spectrum* **25**(1): 1–23.

Smith, T. B. and Pyle, J. M. (2004). An effective algorithm for project scheduling with arbitrary temporal constraints, *AAAI'04 Proceedings of the 19th national conference on Artifical intelligence*, pp. 544–549.

Stahlbock, R. and Voss, S. (2008). Operations research at container terminals: a literature update, *OR Spectrum* **30**(1): 1–52.

Steenken, D., Voss, S. and Stahlbock, R. (2004). Container terminal operation and operations research - a classification and literature review, *OR Spectrum* **26**(1): 3–49.

Tong, C., Lau, H. and Lim, A. (1999). Ant colony optimization for the ship berthing problem, *in* P. Thiagarajan and R. Yap (eds), *ASIAN '99 Proceedings of the 5th Asian Computing Science Conference on Advances in Computer Science*, Vol. 1742 of *Lecture Notes in Computer Science*, Springer, pp. 789–789.

Vacca, I. (2011). *Container terminal management: integrated models and large-scale optimization algorithms*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne.

Zhou, P.-f. and Kang, H.-g. (2008). Study on berth and quay-crane allocation under stochastic environments in container terminal, *Systems Engineering - Theory & Practice* **28**(1): 161 – 169.