IIE Transactions

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uiie20

# Integrated Storage Space Allocation and Ship Scheduling Problem in Bulk Cargo Terminals

Lixin Tang[a], Defeng Sun[a] & Jiyin Liu[b]

[a] Institute of Industrial Engineering & Logistics Optimization, Northeastern University, Shenyang, 110819, P. R. China

[b] School of Business and Economics, Loughborough University, Leicestershire, LE11 3TU, UK

Accepted author version posted online: 29 Jul 2015.

To cite this article: Lixin Tang, Defeng Sun & Jiyin Liu (2015): Integrated Storage Space Allocation and Ship Scheduling Problem in Bulk Cargo Terminals, IIE Transactions

To link to this article: http://dx.doi.org/10.1080/0740817X.2015.1063791

Disclaimer: This is a version of an unedited manuscript that has been accepted for publication. As a service to authors and researchers we are providing this version of the accepted manuscript (AM). Copyediting, typesetting, and review of the resulting proof will be undertaken on this manuscript before final publication of the Version of Record (VoR). During production and pre-press, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal relate to this version also.

# Integrated Storage Space Allocation and Ship Scheduling Problem in Bulk Cargo Terminals

LIXIN TANG[1*], DEFENG SUN[1] and JIYIN LIU[2]

[1] *Institute of Industrial Engineering & Logistics Optimization, Northeastern University, Shenyang, 110819, P. R. China*

[2] *School of Business and Economics, Loughborough University, Leicestershire, LE11 3TU, UK*

## Abstract

This study is motivated by the practice of large iron and steel companies, which have steady and heavy demands for bulk raw materials such as iron ore, coal, limestone and so on. These materials are usually transported to a bulk cargo terminal by ships (or to a station by trains). Once discharged, they are moved to and stored in bulk material stock yard, waiting for retrieval for production use. Efficient storage space allocation and ship scheduling are critical to achieve high space utilization, low material loss and low transportation costs. In this paper, we study the integrated storage space allocation and ship scheduling problem (IASP) in the bulk cargo terminal. Our problem is different from other associated problems due to the special way that the materials are transported and stored. A novel mixed-integer programming model (MILP) is developed, and then solved by Benders decomposition algorithm, which is enhanced by the use of various valid inequalities, combinatorial Benders cuts, variable reduction tests and an iterative heuristic procedure. Computational results indicate that the proposed solution method is much more efficient than the standard solution software CPLEX.

***Keywords*:** Benders decomposition; Integer programming; Valid inequality; Variable reduction tests; Heuristics.

* Corresponding author. E-mail: qhjytlx@mail.neu.edu.cn .

## 1 Introduction

This paper addresses the integrated bulk material storage space allocation and ship scheduling problem arising in the bulk raw material ports of large iron and steel companies. An iron and steel company usually consumes a large quantity of bulk raw materials such as ore and coal which are frequently moved in and out the material stockyard. With the increasing steel production, the material stock yards have become scarce resources in many iron and steel companies. Therefore, it is essential to schedule the ship unloading activities and to effectively allocate storage spaces to unloaded materials.

The material unloading and storage operation decisions determine the allocation costs related to distribution operations and material mixing losses, etc., in the storage yard as well as the scheduling costs associated with unloading sequence of the ships at the berth (or trains in the unloading station). Shipping companies usually charge for the tardiness if the unloading of a ship cannot be completed before the scheduled departure time. Therefore effective storage space allocation and ship scheduling will help iron and steel companies reduce cost and improve profitability.

Figure 1 is a schematic diagram showing the logistics structure and the bulk material flows in a typical iron and steel company. The materials are usually unloaded in the unloading area (berth for example) from ships one by one and then transported to the material stock yard through a complicated belt conveyor network and stored in the form of piles. When needed, the stored materials are collected from the material stock yard by reclaimers and transported to the production facilities or customers, usually through a belt conveyor network too. It is worth noting that for the same material, the unloading operation in the berth and the storing operation in the material stock yard are performed simultaneously.

A material stock yard consists of several stock rows, each of which is a hundreds-of-meters-long strip area capable of storing various materials in the form of piles (as shown in Fig.2). At any particular time, some parts of a row may be occupied by materials stored earlier while other parts (called empty fields) are still available for allocation to new materials. Depending on its length, one empty field may be used to store several piles of different materials if it has enough capacity. Notice that different empty fields may have different lengths. The empty fields may have some residuals left by the materials that were previously stored there. When storing a new material in an empty field, the material will inevitably have some mixing with the residuals at the bottom of the pile. If the residuals are not of the same material, a thin bottom layer of the current material cannot be used as quality material, resulting in material loss.

Figure 2 shows the top and side views of a stock row. We can easily tell the position and the length of each material pile or each empty field in the row from the coordinate axis along the length of the row. As can be seen from figure 2, the width of each material pile is almost the width of the stock row (leaving only narrow margins on the sides). Different material piles in a row must be separated from each other by at least a given safety distance to prevent mixing. The physical characteristics of the materials and the amounts stored determine the shapes and lengths of the piles. Since each material has its fixed bulk density (therefore fixed pile height), we can easily calculate the required storage pile length for a given quantity of a given material. The calculations also take the slope into consideration (see Figure 2b). The slope at both ends of each material pile will cause inevitable capacity waste, which is dependent on the material's type and the width of the allocated stock row, since the width of the row and the slop of the material can affect the amount of the material piled in unit length.

Field 3 in Fig.2 is initially occupied by material 4 at the beginning of the planning horizon, but at a certain time point in the planning horizon, material 4 will be retrieved from the stock row and field 3 will be cleared and become available. Therefore, if a material is allocated to field 3, its unloading and storing operations cannot be started until this available time.

If a material is allocated to some part of an empty field, then the rest part can be seen as a new field and thus available for future use. Moreover, as shown in Fig.2, if some space on the right side of field 2 has not been allocated yet when field 3 is cleared up, then the right part of field 2 together with field 3 will form a longer consecutive new field. This will be useful for those material piles of large quantity, each of which needs long consecutive storage space.

The same type of material arriving with different ships/trains is considered as different materials for traceability and due to their difference in moisture etc. One ship is considered to carry only one material. This is in line with the practical situation we encountered in these bulk cargo terminals of large iron and steel companies.

In bulk cargo terminals, it is important to consider the status of the storage space utilization and the tardiness cost rates of different ships when making unloading sequence decisions. In this case, first-come-first-serve is usually not an optimal policy. The ships do not have a pre-scheduled unloading time until the integrated unloading and allocation decisions are made by the terminals, but each of them has an arrival time and a due time for departure. The duration between the arrival time and the departure time of a ship is normally much longer than the time needed for unloading, leaving flexibility to decide when the ship is unloaded. Given the status of the material stock yard and the ships arriving/due times in the planning horizon, the integrated storage space allocation and ship scheduling problem is to decide the unloading sequences of the material ships, as well as the stock rows and the exact storage locations for each material so as to minimize the associated costs.

Corresponding to a particular bulk material storage yard (e.g. yard used to store iron ore only), there is a relatively fixed berth position for ship unloading. The ship scheduling part of the problem needs only to determine the sequence for unloading the ships. On the other hand, because the unloading operation and the storage operation need to be done simultaneously, the scheduling of ships is closely related to the material storage allocation. Therefore, although there have been many previous studies on berth allocation to ships, the results are not applicable to our problem. Material storage allocation is the major type of decision in our problem. Although there has been plenty of papers focusing on the problems in other production and logistics stages (e.g. Tang et al. 2014, Kim et al. 2011), little research has been done about the bulk raw material production and logistics, especially the bulk raw material storage space allocation problem. Kim et al. (2009) formulated an integer programming model for a bulk material stock yard allocation problem to allocate materials to each available yard, and solved the model using commercial software. The problem is a tactical-level problem faced at the design stage or for longer term planning. Ago et al. (2007) modeled the yard allocation problem and the routing problem simultaneously, but did not consider the actual shapes of the material piles nor the clearance distances between different materials stored in the same storage yard. What was called a yard in these two studies is actually a stock row as shown in Fig. 2. The models in these studies did not consider the exact location of each material in the stock rows, and the available rows were considered all empty before the allocation. In the operational level problem that we study, it is important to consider the spaces already occupied and the exact locations of the available fields in each row. The exact locations for each material to be stored must be specified in the solution so as to provide practical guidance to the actual yard operations.

There have been some similar studies on the space allocation problem encountered in container terminals. Zhang et al. (2003) solved the container storage space allocation problem in container terminals using a rolling-horizon approach. For each planning horizon, the problem was decomposed into two levels, the storage space allocation level and the exact location assignment level. Kim et al. (2000) considered the container storage location problem in the operational level, and solved it using dynamic programming and a decision tree heuristic. Bruzzone and Signorile (1998) combined simulation and genetic algorithms to determine the storage clusters of containers (and the berth allocation) of vessels. Tang et al. (2015) improve the existing static model for container stacking and reshuffling problem in stock yards of container terminals, and develop a discrete event simulation model and five effective heuristics which can together well adapt dynamic environment with arrivals and retrievals of containers. There has also been some research on allocation of specific storage locations to containers in container terminals, e.g., Kim and Hong (2006) and Wan et al. (2009). However, our problem has its special characteristics that are different from those for the container space allocation. The bulk materials do not have a fixed shape and the available fields have different lengths and different available times.

Our major contribution in this paper is to develop a novel mixed integer programming model to first address the practical integrated problem of storage space allocation and ship scheduling in the bulk cargo terminal, and then propose an efficient Benders-based solution procedure to solve it. By investigating deeply into the problem structure, we develop various accelerating strategies which are embedded into the basic Benders algorithm for quick convergence. These strategies include (1) Benders decomposition-based valid inequalities which take the form similar to Benders cuts, (2) effective Combinatorial Benders (CB) cuts which can work even when the Benders subproblem is feasible, (3) several variable reduction tests which can significantly lower the computational burden of the Benders master problem, and (4) a Benders-based heuristic procedure for the priori cuts and variable reduction tests. In order to verify the efficiency of the solution procedure, computational experiments on both the real-world data and random-generated instances are performed.

The rest of the article is organized as follows. The next section describes the bulk material storage space allocation and ship scheduling problem in detail. An integer programming formulation of the problem is developed in Section 3. Section 4 presents an improved Benders decomposition approach to solving the model. In Section 5, we present computational results illustrating the performance of our solution algorithms. Finally, conclusions are drawn in Section 6.

## 2 Problem definition

At any time of the whole planning horizon, some parts of a stock row may be occupied leaving several fields available for use. As the unloading and storage operations goes on, available fields keep changing. Moreover, there may be different residuals left at different parts of a field, resulting in a piecewise cost function of material loss.

To simplify the problem formulation, we view each stock row as a series of unit storage spaces, called slots, each of which is 5 meters long (viz. the length of the minimum safety distance) and can be used to store at most one material. Because some parts of a stock row are occupied by materials that will not be retrieved during the whole planning horizon, they cannot be allocated to store incoming materials in the period and so we do not need to consider their actual size. Instead, we only need to consider an unavailable part as a border slot (BS) separating two adjacent available fields on its two sides, as shown in Figure 3. With this treatment, each row can be viewed as a sequence of fields with border slots between some adjacent fields. The fields in the row can then be marked along the length axis using slot as unit. Note that the lengths of different available fields are often different, and that the lengths of different rows may be different as well. Besides, different slots in the same field may have different available times.

The original rules for space allocation still need to be observed. Each material, brought by one ship, can be allocated to one stock row only, and must be stored in one pile which may occupy several consecutive slots. This is because storing the material in one pile can ensure that the corresponding unloading, transferring and stacking operations can be carried out uninterruptedly. This can also help improve the utilization of the stock yard because it reduces the chance of generating scattered smaller available fields. Moreover, storing a material in one pile makes it easier to track and manage the material since the material transported in a ship has a unique batch number. Between two piles of materials, there must be at least one slot to keep the safety distance. Given a certain material, the maximum amount of this material that can be allocated to any slot in the given stock row is a constant due to the fixed bulk density mentioned earlier. During the whole planning horizon (usually 2 days considering availability of the ship arriving information), the newly allocated materials will need to first go through a checking process (for example, chemical components analysis tests) before use, and such a process normally last for several days. Therefore, once the material is allocated, it will not be retrieved from the storage yard during the same planning horizon. Based on this realistic situation, any available slot cannot be allocated to more than one incoming material during the same planning horizon.

Figure 4 shows an example allocation of 5 slots to a material in a row of 9 slots. Slots 1 to 5 are allocated uninterruptedly and comprise a material pile, leaving slots 6 to 9 as an available field for future use. For the allocated material pile, slots 1 and 5 are called the start slot (SS) and the end slot

(ES), respectively, of the material pile and slot 7 is the SS of the new field. If there are only two slots between two existing material piles, this two slot space will not be considered as an available field because it cannot be used to accommodate a new material pile due to the safety distance needed. It is worth to note that the first and the last slots of each storage row can be used to store materials since there is a natural border before the first slot and after the last one, and so no border slot is needed at the two ends of the row.

An available storage slot may have been used previously to store a material, and has some residuals of that material even after the clearance operation. As noted earlier, when storing a new material in the slot, there will be inevitably some mixing with the residuals at the bottom. Since the quality of the bottom layer of the material is affected, this can be considered as a cost. Obviously, this cost is dependent on the differences between the chemical and physical properties of the two materials.

The material stock yard is connected with the unloading area through conveyer belts, and so the unloading and storing operations are performed simultaneously. The tardiness penalty cost charged by the shipping companies is another significant factor which the managers need to consider when making storage allocation and ship scheduling decisions.

Our objective for allocating storage slots to the materials and scheduling the ships is to minimize the overall cost consisting of all the above mentioned. According to the real-world data, we assume that there are plenty of empty fields to store the incoming materials in the whole planning horizion, which means that the problem we consider is always feasible.

# 3 Model Formulation

## 3.1 Notations

In this section we formulate the integrated storage space allocation and ship scheduling problem as a mixed integer linear programming model (MILP). To present the model, we first define some notations.

*Sets and indices*

$I$ set of incoming materials to be allocated. They are indexed from 1 to $|I|$. This index is also used as ship index since one ship carries only one material.

$R$ set of stock rows that are indexed from 1 to $|R|$.

$S_r$ set of storage slots in row $r$ that are indexed from 1 to $|S_r|$.

*Parameters*

$f_{isr}$ penalty cost (due to the mixing of residuals at the bottom) and distribution cost of storing material $i$ into slot $s$ of stock row $r$.

$h_i$ unit tardiness cost of the ship carrying material $i$.

$d_i$ the planned departure time of the ship carrying material $i$.

$a_i$ the arriving time of the ship carrying material $i$.

$p_i$ the operation time needed to unload and store material $i$.

$b_{ij}$ setup time between the operations from material $i$ to $j$, $i \neq j$ associated with routine adjustment of the belt conveyers.

$w_{ir}$ capacity waste at the ends of the piles (due to slope) when allocating material $i$ to row $r$.

$q_{ir}$ maximum quantity of material $i$ that can be stored into each slot of row $r$.

$q_{i0}$ total quantity of material $i$ to be allocated.

$e_{rs}$ the earliest available time of slot *s* in row *r*.

$k_{rs}$ parameter indicating whether slot *s* is the BS in stock row *r* or not (equals 0 when true, and 1 otherwise).

$M$ a positive number that practically serves as infinity for the problem.

$|S_r|+1$ a virtual slot in row *r* outside $|S_r|$ to indicate the end of the row.

### *Decision variables*

$t_i$ the starting time of material *i*'s unloading operation.

$C_i$ the tardy time of material *i*'s unloading operation.

$u_{ij}$ one if material *j* is unloaded after material *i*, zero otherwise.

$y_{irs}$ one if material *i* is allocated to slot *s* of row *r*, zero otherwise. As the virtual slot $|S_r|+1$ cannot be used to store any material, we define $y_{ir,|S_r|+1} \equiv 0$ for any *i* and *r*.

$x_{irs}$ one if slot *s* is the ES for material *i* in stock row *r*, zero otherwise.

$z_{ir}$ one if material *i* is allocated to stock row *r*, zero otherwise.

## 3.2 Model formulation

Using the notations above, we can formulate the MILP as follows:

$$\text{(MILP) Minimize } \sum_{i \in I} \sum_{r \in R} \sum_{s \in S_r} f_{irs} y_{irs} + \sum_{i \in I} h_i C_i \qquad (1)$$

Subject to

$$C_i \geq t_i + p_i - d_i \qquad \forall i \in I \qquad (2)$$

$$t_j - t_i + M\left(1 - u_{ij}\right) \geq p_i + b_{ij} \qquad \forall i, j \in I,\ i \neq j \qquad (3)$$

$$u_{ij} + u_{ji} = 1 \qquad \forall i, j \in I,\ i \neq j \qquad (4)$$

$$t_i \geq e_{rs} y_{irs} \qquad \forall i \in I, r \in R, s \in S_r \qquad (5)$$

$$t_i \geq a_i \qquad \forall i \in I \qquad (6)$$

$$\sum_{r \in R} z_{ir} = 1 \qquad \forall i \in I \qquad (7)$$

$$\sum_{s \in S_r} q_{ir} y_{irs} - w_{ir} z_{ir} \geq q_{i0} z_{ir} \qquad \forall i \in I, r \in R \qquad (8)$$

$$\sum_{i \in I} y_{irs} \leq k_{rs} \qquad \forall r \in R, s \in S_r \qquad (9)$$

$$\sum_{j \in I \setminus \{i\}} y_{jr,s+1} \leq 1 - y_{irs} \qquad \forall i \in I, r \in R, s \in S_r \qquad (10)$$

$$y_{irs} - y_{ir,s+1} \leq x_{irs} \qquad \forall i \in I, r \in R, s \in S_r \qquad (11)$$

$$\sum_{s \in S_r} x_{irs} \leq z_{ir} \qquad \forall i \in I, r \in R \qquad (12)$$

$$t_i, C_i \geq 0 \qquad \forall i \in I \qquad (13)$$

$$x_{irs}, y_{irs}, z_{ir}, u_{ij} \in \{0,1\} \qquad \forall i \in I, j \in I, r \in R, s \in S_r \qquad (14)$$

The objective (1) is to minimize the total cost associated with the allocation. The first term represents the residuals mixing costs and distribution costs. The second term represents the tardiness penalty costs charged by the ships.

Constraints (2) are used to calculate the tardiness time of each material. Constraints (3) ensure that the unloading of a material can start only when the unloading of any material scheduled before it has completed and a setup is done. For any two materials, constraints (4) require that one of them must be unloaded after another. Constraints (5) ensure that, if a material is allocated to a slot, the starting time of its unloading cannot be earlier than the earliest available time of the slot. Constraints (6) ensure that each material cannot be unloaded until the ship carrying this material arrives. Constraints (7) guarantee that each material can be allocated to only one stock row. Constraint set (8) ensures that sufficient slots are allocated for each material. Constraints (9) guarantee that each slot can be used to allocate at most one incoming material in the planning horizon. Constraints (10) are the safety distance constraints. Constraint set (11) identifies (the ES of) the piles of materials in each row. Constraint set (12) ensures that only one pile is stored on the chosen stock row for each material. Constraints (13) and (14) are the binary and non-negativity constraints.

We propose to solve this problem using the Benders decomposition method, which could decompose it into two easier problems. This decomposition method could also inspire us to develop various accelerating strategies associated with the problem structure (e.g., quick ways to carry out variable reduction tests and generation of Combinatorial Benders cuts, illustrated in later sections).

## 4 Benders decomposition based solution approach

Benders Decomposition (BD) is an effective method applicable to mixed-integer programs (Benders, 1962), based on the partition and delayed constraint generation ideas. It decomposes the original problem into two simpler ones: an integer master problem and a linear subproblem, which are solved in an iterative fashion by utilizing the solution of one in the other. The master problem actually behaves as a relaxation of the original problem and involves all the integer variables and one continuous auxiliary variable to incorporate the information transferred from the subproblem. Using the fixed integer variable values obtained from the master problem solution as input parameters, a dual subproblem is constructed and its solution provides the means to construct a Benders cut that includes the information of the subproblem. The Benders cut is then added to the master problem in the next iteration. The Benders cut excludes the solution just obtained in the last master problem. Therefore, each solution of the master problem must satisfy all the Benders cuts generated so far to avoid repetition. Solving the master problem and the dual subproblem can provide a lower bound and an upper bound for the overall problem, respectively. The master problem and the subproblem are solved iteratively in this fashion until an optimal solution to the original problem is obtained.

Most studies on Benders focus on how to improve the quality of Benders cuts, or how to solve the integer master problem quickly. Magnanti and Wong (1981) first define a cut as Pareto-optimal if no other cut dominates it, which can significantly improve Benders if applied to problems with degenerate subproblem. Tang et al. (2013) improve this technique by a high density Pareto cut generation strategy. Saharidis et al. (2011) present a Covering Cut Bundle strategy to accelerate Benders decomposition algorithm by generating a bundle of cuts in order to cover all the decision variables of the master problem.

Solving the integer master problem to optimal could help the subproblem to generate the most powerful cut. However it is time consuming. In fact, if any feasible solution (even if the solution of the LP relaxation) of the master problem is passed to the subproblem, the cuts generated are still valid for the integer programming problem (McDaniel and Devine, 1977). We refer to this as **Property 1** in the rest of this paper.

### 4.1 Benders reformulation

Let $\bar{y}_{irs}$ and $\bar{u}_{ij}$ represent the given values of the integer variable $y_{irs}$ and $u_{ij}$. Then, for any fixed values $\bar{y}_{irs}$ and $\bar{u}_{ij}$, the Benders subproblem **SP** is given as the following linear program which includes all the original constraints involving the continuous variables:

(**SP**) Minimize $\sum_{i \in I} h_i C_i$ (15)

Subject to

$$C_i - t_i \geq p_i - d_i \qquad \forall i \in I \tag{16}$$

$$t_j - t_i \geq p_i + b_{ij} - M\left(1 - \bar{u}_{ij}\right) \qquad \forall i, j \in I,\ i \neq j \tag{17}$$

$$t_i \geq e_{rs}\bar{y}_{irs} \qquad \forall i \in I, r \in R, s \in S_r \tag{18}$$

$$t_i \geq a_i \qquad \forall i \in I \tag{19}$$

$$C_i, t_i \geq 0 \qquad \forall i \in I \qquad (20)$$

Note that constraint (16)-(18) is the Benders rewritten form of constraint (2), (3) and (5).

Therefore, the dual sub-problem (**DSP**), which is the dual of **SP** can be written as:

(**DSP**) Maximize
$$\sum_{i\in I}(p_i - d_i)\alpha_i + \sum_{i\in I, i\neq j}\sum_{j\in I}\left[p_i + b_{ij} - M\left(1-\bar{u}_{ij}\right)\right]\beta_{ij} + \sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} e_{rs}\bar{y}_{irs}\theta_{irs} + \sum_{i\in I} a_i\varepsilon_i \qquad (21)$$

Subject to

$$\alpha_i \leq h_i \qquad \forall i \in I \qquad (22)$$

$$\sum_{j\in J}\beta_{ji} - \sum_{j\in J}\beta_{ij} - \alpha_i + \varepsilon_i + \sum_{r\in R}\sum_{s\in S_r}\theta_{irs} \leq 0 \qquad \forall i \in I \qquad (23)$$

$$\alpha_i, \beta_{ij}, \theta_{irs}, \varepsilon_i \geq 0 \qquad \forall i \in I, j \in I, r \in R, s \in S_r \qquad (24)$$

Utilizing the above representation of the **DSP** and based on the extreme points and the extreme rays of its polyhedron, we can introduce a variable $\pi$ for the overall tardiness penalty costs and formulate the master problem as:

(**MP**) Minimize
$$\pi + \sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs} y_{irs} \qquad (25)$$

Subject to

constraints (4), (7) - (13)

$$\pi \geq \sum_{i\in I}(p_i - d_i)\alpha'_i + \sum_{i\in I, i\neq j}\sum_{j\in J}\left[p_i + b_{ij} - M\left(1-u_{ij}\right)\right]\beta'_{ij} + \sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} e_{rs} y_{irs}\theta'_{irs} + \sum_{i\in I} a_i\varepsilon'_i \qquad (26)$$

$$0 \geq \sum_{i\in I}(p_i - d_i)\alpha''_i + \sum_{i\in I, i\neq j}\sum_{j\in J}\left[p_i + b_{ij} - M\left(1-u_{ij}\right)\right]\beta''_{ij} + \sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} e_{rs} y_{irs}\theta''_{irs} + \sum_{i\in I} a_i\varepsilon''_i \qquad (27)$$

We can iteratively solve **MP** and **DSP** until an optimal solution to the original problem is obtained. In each iteration a new Benders cut (26) associated with the extreme point or (27) associated with the extreme ray of last **DSP** is added to **MP**.

## 4.2 Valid inequalities

Observe that the iterative algorithm BD is initialized with empty subsets of extreme rays and extreme points, therefore the master problem initially contains only the integrality constraints. As a result, several iterations must be performed before enough information is transferred to the master problem by Benders cuts.

When solving model **MP**, various types of valid inequalities can be added to the formulation. They can improve convergence by helping the master problem to find solutions that are feasible and close to optimal.

In this section, besides two problem-structure-based valid inequalities, we also introduce a new type that we called Benders-based valid inequality, which is derived from the idea of imitating the Benders cuts, to help the master problem find near-optimal solutions.

### 4.2.1 *Valid inequality of unloading sequence*

We find that the absence of constraints (3) in **MP** may result in $u_{ij}$ solution values that cause infeasibility of **SP**. To handle this situation, we investigate the optimal solution structure of $u_{ij}$ and add the following valid inequality into **MP**:

$$\sum_{j\in I\setminus\{i\}} u_{ij} \neq \sum_{j\in I\setminus\{i'\}} u_{i'j} \qquad \forall i\in I, i'\in I\setminus\{i\} \tag{28}$$

This valid inequality is based on the observation that each material (ship) occupies a unique position in the feasible unloading sequence. It can reduce the possibility of having $u_{ij}$ solution values that make **SP** infeasible. Note that (28) is allowable in the standard commercial software CPLEX we used when implementing our solution approach.

### 4.2.2 *Valid inequality based on capacity*

Between two adjacent border slots (or before the first border slot or after the last border slot) in a row is a sequence of consecutive available slots though their available times may be different. If there are no new materials stored, this sequence of slots will all be empty at some time point of the planning horizon, and so can potentially accommodate a material with large quantity. We refer to such a sequence of slots as a potential field.

Considering the allocation structure of material *i*, if a potential field is not long enough to accommodate the quantity of the material, the material cannot be allocated to any slot in this potential field and the corresponding decision variable $y_{irs}$ should be 0. That is

$$\sum_{(r,s)\in G_i} y_{irs} \le 0 \qquad \forall i \in I \tag{29}$$

where $G_i$ represents the set of slots in all the potential fields that are not long enough to allocate material *i*. As the border slots and the quantity of material *i* are known, $G_i$ can be obtained easily in advance.

### 4.2.3 *Benders-based valid inequality*

Given a general mix-integer programming problem MIP:{Min c*x*+d*y* | A*x*+B*y* ≥ a; C*y*≤b; *x*≥0; *y* ∈{0, 1}}, its master problem can be written as MP:{Min d*y*+π | C*y*≤b; π≥CutExp(*y*); *y*∈{0, 1}}, where π≥CutExp(*y*) is the Benders cut expression. Let SP(*y*) represent the optimal objective value of the associated subproblem when the value of *y* is given. As illustrated earlier, in each iteration of the Benders algorithm, MP and d*y*+SP(*y*) provides the LB and UB of MIP respectively, and MIP is solved to optimal when LB=UB.

In this section, we investigate the possibility of developing valid inequalities by imitating the Benders cuts. We call this strategy as Benders-based inequality, which is based on the observation that a Benders cut is constructed as a linear function of Boolean variables.

Before we illustrate the Benders-based valid inequality strategy, we first define expression RL(*y*) to be used later as a linear function of Boolean variables *y*.

**Proposition 1.** If a given expression RL($y$) satisfies SP($y$*)≥RL($y$*) for the optimal solution of MIP (denoted as $y$*), then the cut-like inequality π≥RL($y$) is valid for the master problem MP. We call it the Benders-based valid inequality.

**Proof.** We first construct an auxiliary master problem AMP by adding the Benders-based inequality π≥RL($y$) into the master problem MP. Then we employ a Benders algorithm structure in which we solve AMP and SP iteratively until achieving the optimum. Let $y$^ denote this optimal solution. Note that when reaching this optimal solution, the value of AMP (i.e. d$y$^+π($y$^)) equals the associated upper bound d$y$^+SP($y$^), which is also an upper bound (UB) of MIP. If the Benders-based inequality π≥RL($y$) is not valid for the master problem MP, then $y$^ is not the optimal solution of MIP, that is to say, $y$^ is different from $y$*, then for the current AMP polyhedron we have d$y$*+π($y$*) > d$y$^+π($y$^)=UB since $y$^ is the optimal solution of the current AMP polyhedron. Observe that d$y$*+CutExp($y$*) is a lower bound of MIP (based on **Property 1)**, then we have RL($y$*)>CutExp($y$*) and further d$y$*+π($y$*)=d$y$*+RL($y$*). However, given that SP($y$*)≥RL($y$*), we have d$y$*+SP($y$*) ≥ d$y$*+RL($y$*) = d$y$*+π($y$*)> d$y$^+π($y$^) = UB, which is definitely a contradiction because d$y$*+SP($y$*) is the optimal objective value of MIP (denoted as OPT) and the assumption leads to OPT>UB. Therefore, $y$^ is the optimal solution of MIP, which means that the Benders-based inequality π≥RL($y$) is valid.

Effective RL($y$) expression will prevent the generation of integer solutions far from optimal and further reduce the number of iterations. In our problem, based on (16)-(20), we can easily observe that $h_iC_i \geq h_i(e_{rs}y_{irs}+p_i-d_i)$. Therefore, it is obvious that SP($y$) ≥ RL($y$) = $\sum_i h_i\left(e_{rs}y_{irs}+p_i-d_i\right)$ for any solution $y$. Moreover, based on the problem structure and the observation $h_iC_i \geq h_i(e_{rs}y_{irs}+p_i-d_i)$, the Benders-based valid inequality π≥RL($y$) we established is separable, and can be further converted to the form of

$$\pi \geq \sum_{i \in I} \pi'_i \tag{30}$$

$$\pi'_i \geq h_i \left( e_{rs} y_{irs} + p_i - d_i \right) \qquad \forall i \in I \tag{31}$$

$$\pi'_i \geq h_i \left[ \frac{\pi'_j}{h_j} + b_{ji} + p_i - d_i - M\left(1 - u_{ji}\right) \right] \qquad \forall i \in I, j \in I \setminus \{i\} \tag{32}$$

$$\pi'_i \geq 0 \qquad \forall i \in I \tag{33}$$

Actually, given a problem-structure-based RL(*y*) expression that does not satisfy RL(*y**)≤SP(*y**), this strategy may still accelerate the BD algorithm. As in the proof above, we can solve AMP instead of MP in the first few steps of standard Benders procedure. Since the Benders cuts generated in these few steps is still valid because of **Property 1**, we can probably improve convergence by helping the master problem to find solutions that are close to optimal.

### 4.3 Combinatorial Benders cut

Once the linear subproblem has big-M constraints (like constraint (17) in our problem), the resulting Benders cuts will be weak especially when the subproblem is infeasible. In order to break the infeasibility effectively, Hooker (2000) propose a Benders cuts generating technique based on the idea of *minimal set of inconsistencies*. Codato and Fischetti (2006) further extend it to solve the MIP with special structures using combinatorial Benders cuts (CB cuts), by figuring out the integer variables that need to be changed to break the infeasibility. Cao et al. (2010) apply this strategy to solve an integrated yard truck and yard crane scheduling problem with the help of an auxiliary linear system.

In this paper, by constructing infeasible auxiliary subproblem, we chose to take advantage of combinatorial Benders cuts even when the subproblem **SP** is feasible. That is to say, we chose to figure out the integer variables that need to be changed to help the algorithm get close to the optimal solution. To achieve that, we propose two strategies as follows:

### 4.3.1 *CB-cut generating strategy one*

Similar to Cao et al. (2010), when solving the subproblems **SP** in each Benders iteration, we introduce the following linear system:

$$\textbf{Sub(y,u)} := \begin{cases} \text{constraints}(16)-(20) \\ \sum_{i\in I} h_i C_i \le \text{UB-MPV-}\varepsilon \end{cases} \tag{34}$$

where UB is the incumbent upper bound, MPV is a parameter satisfying MPV$\le \sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs} y^*_{irs}$ , $y^*_{irs}$ is the optimal value of $y_{irs}$ in the optimal solution of MILP, and $\varepsilon$ is a sufficiently small positive value.

Let $C^*_i$ denote the optimal value of variables $C_i$ in **MILP**, it can be easily observed that $\text{MPV+}\sum_{i\in I} h_i C^*_i \le \sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs} y^*_{irs} + \sum_{i\in I} h_i C^*_i \le \text{UB}$, therefore $\sum_{i\in I} h_i C^*_i \le \text{UB-MPV}$, i.e. the optimal value of integer variables in **MILP** should guarantee that the subproblem **SP** satisfies constraints (34). Therefore, we can proceed as follows:

When **Sub(y,u)** is feasible, we can update the incumbent upper bound UB by solving **SP**; otherwise, if **Sub(y,u)** is infeasible, we can look for the *minimal infeasible subsystem* (*MIS*) of **Sub(y,u)**, indexed by $\Omega$, and observe that at least one binary variables involved in it has to be changed to break the infeasibility so as to get close to the optimal solution. This observation can be expressed as combinatorial Benders cut:

$$\sum_{(i,j)\in\Omega_1:\bar{u}_{ij}=0} u_{ij} + \sum_{(i,j)\in\Omega_1:\bar{u}_{ij}=1} \left(1-u_{ij}\right) + \sum_{(i,r,s)\in\Omega_2,\bar{y}_{irs}=0} y_{irs} + \sum_{(i,r,s)\in\Omega_2,\bar{y}_{irs}=1} \left(1-y_{irs}\right) \ge 1 \tag{35}$$

where $\Omega = \Omega_1 \cup \Omega_2$. The subset $\Omega_1$ contains indices of rows corresponding to constraint (17); and the subset $\Omega_2$ contains indices of rows corresponding to constraint (18).

When the subproblem **SP** is infeasible, **MILP** and **Sub(y,u)** are both infeasible, we could look for an *MIS* of **SP** (or **Sub(y,u)**) to generate a CB cut, and $\Omega_2 = \varnothing$ in this case.

Because of the structure of Benders algorithm, we can easily get a proper value of MPV by setting MPV= $\sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs} y_{irs}^1$ , where $y_{irs}^1$ denotes the optimal value of $y_{irs}$ obtained when solving the mater problem **MP** in the first iteration. Observe that $\sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs} y_{irs}^1 \le \sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs} y_{irs}^*$ since only the item $\sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs} y_{irs}$ in the objective function is involved when solving MP in the first iteration.

### 4.3.2 *CB-cut generating strategy two*

Considering the large problem scale of **MP** and the risk of infeasible **SP**, we further give an easier way to generate CB cuts (and initial cuts) by employing a set of auxiliary problems.

In the $k^{\text{th}}$ iteration of the Benders approach, we relax all the binary integer variables in **MP**$^k$, except $u_{ij}$, to [0,1] to get a relaxed master problem **MP**$^k{}_{\text{LP}}$. We first solve **MP**$^k{}_{\text{LP}}$ and obtain the optimal value of variables $y_{irs}$, $z_{ir}$ and $u_{ij}$ in **MP**$^k{}_{\text{LP}}$, denoted as $\hat{y}_{irs}^k, \hat{z}_{ir}^k$ and $\hat{u}_{ij}^k$. Then solving the **SP** involving $\hat{y}_{irs}^k, \hat{z}_{ir}^k$ and $\hat{u}_{ij}^k$, we will encounter two cases:

Case 1: If the **SP** is infeasible, we can generate a CB cut through investigating the *MIS* of **SP**, indexed by $\Omega_3$. This CB cut can be illustrated as:

$$\sum_{(i,j)\in\Omega_3:\hat{u}_{ij}=0} u_{ij} + \sum_{(i,j)\in\Omega_3:\hat{u}_{ij}=1} \left(1-u_{ij}\right) \ge 1 \tag{36}$$

**Proposition 2**. If the **SP** involving $\hat{y}_{irs}^k$ , $\hat{z}_{ir}^k$ and $\hat{u}_{ij}^k$ is infeasible, then the resulting combinatorial Benders cut (36) is valid for **MILP**.

**Proof.** We know **SP** is infeasible for any solution of **MP**$^k$ that satisfying $u_{ij}^k = \hat{u}_{ij}^k$ . It is easy to check that (36) only excludes the exact $u_{ij}$ values that make the problem infeasible but allows any other combination of $u_{ij}$ values. Therefore (36) is a valid cut.

Case 2: If the **SP** is feasible, besides the general cut generating procedure, we may further generate an additional CB cut with the help of an auxiliary linear system. Relax all the integer variables in **MILP** except $u_{ij}$ to get a relaxation problem **MILP**$_{\mathrm{LP}}$. Fix the variables $u_{ij} = \hat{u}_{ij}^{k}$ in **MILP**$_{\mathrm{LP}}$ to get an auxiliary problem **A_MILP**$_{\mathrm{LP}}$, let $v$(**A_MILP**$_{\mathrm{LP}}$) denote the optimal solution value of **A_MILP**$_{\mathrm{LP}}$. We introduce the following linear system:

$$\textbf{MILP\_CB} := \begin{cases} \textbf{A\_MILP}_{\mathrm{LP}} \\ v(\textbf{A\_MILP}_{\mathrm{LP}}) \le \mathrm{UB}\text{-}\varepsilon \end{cases} \tag{37}$$

If the **MILP_CB** is infeasible, we can generate a CB cut through investigating the *MIS* of **MILP_CB**, indexed by $\Omega_4$. This CB cut can be illustrated as:

$$\sum_{(i,j)\in\Omega_4:\hat{u}_{ij}=0} u_{ij} + \sum_{(i,j)\in\Omega_4:\hat{u}_{ij}=1} \left(1-u_{ij}\right) \ge 1 \tag{38}$$

**Proposition 3**. If the **MILP_CB** is infeasible, the resulting combinatorial Benders cut (38) is valid for **MILP**.

**Proof.** This is based on the fact that $v$(A_**MILP**$_{\mathrm{LP}}$) is the lower bound of **MILP** for any solution that satisfying $u_{ij} = \hat{u}_{ij}^{k}$. Therefore, if **MILP_CB** is infeasible, then at least one binary variables involved in the *MIS* $\Omega_4$ has to be changed to break the infeasibility so as to get close to the optimal solution of **MILP**.

Consequently, in the $k^{\mathrm{th}}$ iteration, before solving **MP**$^{k}$, we could first iteratively solve **MP**$^{k}{}_{\mathrm{LP}}$ and **SP**, until **SP** is feasible and **MILP_CB** is feasible. During this iterative procedure in the $k^{\mathrm{th}}$ iteration of Benders approach, a set of CB-cuts and traditional Benders cuts are generated, and could further restrain the solution space of **MP**$^{k}$.

Moreover, based on **Property 1**, before starting the Benders procedure, we could generate a set of initial cuts by solving the easier relaxation problem **MILP**$_{\text{LP}}$ in Benders style involving **MP**$_{\text{LP}}$ and **SP**. The CB-cut generating strategies illustrated in section 4.3.2 can still fit well in this initial-cut-generating procedure.

## 4.4 Variable reduction tests

The number of variables and constraints is very high in large scale instances, making the master problem a tough computational burden. However, we can figure out two efficient strategies to handle this situation, one is to generate a set of initial Benders cut by solving a series of much easier integer master problems as illustrated earlier, the other is to reduce the number of variables using some reduction tests (e.g. Contreras et al. 2011 for hub location problem).

In this section, we develop three different reduction tests capable of eliminating variables that are known not to appear in an optimal solution. The basic idea of most of these reduction tests comes from the observation of a contradiction. Assuming that a Boolean variable takes the value of 1 in the final optimal solution, if the optimal objective value of the problem under this assumption turns out to be greater than the upper bound, contradiction happens, and therefore this variable should be fixed to 0 in the optimal solution. Let UB denote an upper bound of **MILP**.

**Proposition 4**. Given a parameter MPV2 satisfying MPV2≤ $\sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs} y^{*}_{irs}$ and a parameter $E_{irs}$ satisfying $E_{irs} = e_{rs} + p_i - d_i$, if we have MPV2 + $E_{irs}$ > UB, then $y_{irs}$=0 in the optimal solution of **MILP**.

**Proof.** Because of constraints (2) and (5), we have $C_i \geq e_{rs} y_{irs} + p_i - d_i$ . Then we can easily observe the fact that MPV2 + $E_{irs}$ is a lower bound of **MILP** if $y_{irs}$=1. Therefore, if $v$(**MILP**$_{\text{LP}}$) + $E_{irs}$ > UB, then $y_{irs}$=0 in the optimal solution of **MILP**.

We can obtain the value of MPV2 by choosing the larger one of $\sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs}y^1_{irs}$ and $v$(**MILP**$_{\text{LP}}$) - $\sum_{i\in I} h_i C_i$ , where $C_i$ is the optimal value of $C_i$ in **MILP$_{\text{LP}}$**.

**Proposition 5**. Let ***IR*** be a subset of {($i,r$) | $i\in I$; $r\in R$ }, and $v$[**MP**$^k$(***IR***)] be the optimal value of the **MP**$^k$ with additional constraint $\sum_{(i,r)\in IR} z_{ir} \geq 1$. If $v$[**MP**$^k$(***IR***)] >UB, then $z_{ir}$=0 for each ($i,r$)∈***IR*** in the optimal solution of **MILP** (also the corresponding $y_{irs}$=0, $\forall s \in S_r$ according to constraints (11) and (12)).

**Proof.** It is certain that $v$[**MP**$^k$(***IR***)] is a lower bound of the optimal value of any **MP** with constraint $z_{ir}$=1 for any ($i,r$) ∈***IR***. Therefore, if $v$[**MP**$^k$(***IR***)] >UB, then $z_{ir}$=0 in the optimal solution of **MILP**.

Consequently, we can further observe an easier reduction test inspired by **Proposition 5**:

**Proposition 6**. Let $v$[**MP**$^k_{\text{LP}}$(***IR***)] be the optimal value of **MP**$^k_{\text{LP}}$ requiring $\sum_{(i,r)\in IR} z_{ir} \geq 1$ for a subset ***IR*** of {($i,r$) | $i\in I$; $r\in R$ }. If $v$[**MP**$^k_{\text{LP}}$ (***IR***)] >UB, then $z_{ir}$=0 for each ($i,r$)∈***IR*** in the optimal solution of **MILP**.

**Proof.** Similar to **Proposition 5**, this proposition is based on the fact that $v$[**MP$^k_{\text{LP}}$** (***IR***)] is a lower bound of any **MP** requiring $z_{ir}$=1 for any ($i,r$)∈***IR***.

Moreover, once the reduction tests confirm that a $y_{irs}$ variable must be 0 in the optimal solution, we can view slot ($r,s$) as a special border slot only effective for material $i$. Such special border slots will make some previously feasible potential fields for material $i$ become infeasible. Based on this we can update the infeasible slot set for material $i$, $G_i$, as defined in section 4.2.2. We refer to this observation as **Proposition 7**. This is in fact a ***further reduction test*** since we will be able to fix more $y$ variables according to (29).

Variable reduction tests in this section could reduce the number of integer variables, and can be applied in every step of Benders procedure. The valid inequalities mentioned in section 4.2 can help the master problem find solutions that are feasible and close to optimal, especially in the first few iterations, resulting in better lower bound. The combinatorial cut strategy in section 4.3 aims at generating effective cuts in fewer iterations and having less computational burden, to restrict the solution space of master problem.

Furthermore, ingenious combination of these strategies will bring about faster convergence consuming less computation time. In the next section, we will present a Benders-based solution procedure, which includes all the above strategies as well as another useful observation.

### 4.5 A Benders-based solution procedure

After investigating the problem structure in the last section, we now outline the whole procedure of the proposed Benders-based solution algorithm that can make good use of all the accelerating strategies illustrated earlier and quickly solve the problem to optimal.

There are mainly three layers in the procedure. (i) The first layer is a Benders based iterative heuristic procedure containing Steps 1-6, in which we make use of the LP relaxation and the structural properties of the master problem to generate effective Benders cuts and further Combinatorial Benders cuts illustrated in section 4.3; and meanwhile, in every iteration of this layer we could utilize the variable reduction tests illustrated in Proposition 4 and 7 to reduce the number of integer variables. (ii) The second layer contains Step 7 which is based on the variable reduction test illustrated in Proposition 6, aiming at further reducing the number of integer variables. (iii) The third layer is the typical Benders procedure illustrated in Step 8, which gives the optimal solution value. Note that in this layer, with the variable reduction results and the various cuts generated in the first two layers, we can easily get the optimal solution in a much shorter computational time.

Utilizing the observation that the values of $z_{ir}$ in the optimal solution of **MP**$_{\mathrm{LP}}$ indicate the preferable rows for each material in **MP**, we established this Benders-based solution procedure. The whole solution procedure is illustrated as follows (see Table A in the Appendix for index of referred optimization programs):

***Step 0***: Initialization. $I_1$=$I$, $k$=1.

***Step 1***: In the $k^{th}$ iteration of the Benders-based iterative heuristic procedure, first solve **MP**$^{k}_{\mathrm{LP}}$ to optimal and obtain the values $\hat{y}^k_{irs}$, $\hat{z}^k_{ir}$ and $\hat{u}^k_{ij}$. Then solve **SP** employing $\hat{y}^k_{irs}$, $\hat{z}^k_{ir}$ and $\hat{u}^k_{ij}$. If **SP** is feasible, solve **MILP_CB**, and go to *Step 2* if **MILP_CB** is also feasible. If **SP** is infeasible, or a feasible **SP** leading to an infeasible **MILP_CB**, generate a traditional Benders cut and/or a CB cut as illustrated in section 4.3 (see *CB-cut generating strategy two*), and add these cuts into the cut set **BendersCutSet_Initial** and **MP**$^{k}_{\mathrm{LP}}$, repeat *Step 1*.

***Step 2***: Fix variables $u_{ij}=\hat{u}^k_{ij}$ in **MP**$^k$. Check the values of $\hat{z}^k_{ir}$ for each material $i \in I_1$. If $\hat{z}^k_{ir}$=0 then fix variables $z_{ir}$=0 as well as the associated $y_{irs}$=0 in **MP**$^k$. Then we obtain a reduced master problem **MP**$^k$(*reduced*).

***Step 3***: Solve **MP**$^k$(*reduced*). If it is feasible, update the dummy lower bound LB_D = $v$[**MP**$^k$(*reduced*)], and turn to *Step 5*; otherwise, proceed to *Step 4*.

***Step 4***: Find the material $i$ in $I_1$ that gives the largest value of $\min\left\{\left(\lceil \hat{z}^k_{ir} \rceil - \hat{z}^k_{ir}\right) \middle| r \in R\right\}$ and delete it from $I_1$. Go to *Step 2*.

***Step 5***: Solve **SP** to get a traditional Benders cut and a CB cut (if possible, see *CB-cut generating strategy one* illustrated in section 4.3), as well as an upper bound of **MILP**. Add these initial cuts into the cut set **BendersCutSet_Initial**. Update the incumbent upper bound (UB) of **MILP** and do the reduction test illustrated in **Proposition 4** as well as the further reduction test illustrated in **Proposition 7** to reduce the number of variables.

***Step 6***: If LB_D = UB, turn to *Step 7*; otherwise, *k*=*k*+1, turn to *Step 1*.

***Step 7***: Start ***IR***={(*i*,*r*)|*i*∈*I*,*r*∈*R*}, the full set of ships and rows. Solve the following auxiliary master problem (with some variables fixed based on the reduction tests):

**[MP**$_{\text{LP}}$**(*IR*)]** Minimize
$$\pi + \sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs}y_{irs} \tag{39}$$

Subject to

constraints (4), (7)-(13), (26)-(33)

cuts in **BendersCutSet_Initial**

$$\sum_{(i,r)\in IR} z_{ir} \geq 1 \tag{40}$$

$$\pi + \sum_{i\in I}\sum_{r\in R}\sum_{s\in S_r} f_{irs}y_{irs} < \text{UB} \tag{41}$$

$$x_{irs}, y_{irs}, z_{ir} \in [0,1], u_{ij} \in \{0,1\} \qquad \forall i\in I, j\in I, r\in R, s\in S_r \tag{42}$$

If **MP**$_{\text{LP}}$(***IR***) is feasible, remove from ***IR*** all the (*i*,*r*) with $z_{ir}$>0 in the optimal solution of **MP**$_{\text{LP}}$(***IR***), then solve **SP** and **MP**$_{\text{LP}}$(***IR***) iteratively until **MP**$_{\text{LP}}$(***IR***) is infeasible or *IR*=$\varnothing$. Perform the reduction test illustrated in **Proposition 6** to reduce the number of variables.

***Step 8***: Solve **MILP** to optimal in Benders style (with all the cuts generated earlier and all the variable reduction results), stop. Note that CB-cut generating strategy one illustrated in section 4.3 and the reduction tests illustrated in **Propositions 4**&**7** are also applied in the procedure of this step.

# 5 Computational results

To evaluate the computational efficiency of the proposed solution approach, we test it on the real-world data obtained from an iron and steel corporation in China. Furthermore, in order to investigate the efficiency of the proposed approach in different problem scales, we generate more test instances that reflect real application and use them in further test. To benchmark the performance of the Benders algorithm, the MILP models for the instances are solved using CPLEX 12.6. CPLEX 12.6 is also employed to solve the Benders master problem and subproblem. All the computational experiments are performed on a computer with Intel Core i3-2350 2.30 GHz CPU and 4GB RAM.

## 5.1 Results of real-world data

We first test the practical data obtained from a bulk material stock yard of an iron and steel corporation, which has 7 stock rows and has around 15 ships to be unloaded on average in each planning horizon. The average length of a stock row is 800 meters, therefore the maximum numbers of slots in one stock row is 160. We solve ten real-world instances and summarize the CPU time (hour: minute: second) in Table 1. In the table, B represents the basic Benders approach, B+C represents the Benders approach with the combinatorial Benders cut in section 4.2, B+C+V represents B+C with valid equalities (28)-(33) and BA represents the proposed Benders-based approach illustrated in section 4.5. We observe that reducing the optimality gap below 1% requires a massive computational effort. Therefore in order to avoid high computational time, a stopping criterion of a 1% optimality gap is used. A time limit of 12 CPU hours is imposed for any method to solve an instance.

The results reported in Table 1 indicate that solving the model directly by CPLEX takes very long but the gap requirement has been achieved within the 12 hour time limit. The fact that it stops when the gap is equal to or very close to the required gap demonstrates that the gap reduces very slowly in the solution process.  The traditional Benders approach perform poorly due to the large problem scale of the master problem and high infeasible risk of the subproblem in each iteration. For each of the problem instances, it reaches the 12 hour time limit when the optimality gap is still very large.  The accelerating strategies presented in section 4 can accelerate the Benders approach significantly.

We can see from Table 1 that every strategy can contribute to shortening the computation time. We also measure the numbers of combinatorial Benders cuts appeared in different approaches. Table 2 presents the average number of these cuts as well as the average of first lower bounds (and the corresponding relative difference compared to standard Benders) obtained when basic Benders procedure begins in each approach. Note that when we do the measurement in approach B+C and B+C+V, we also take into account the effect of the initial-cut-generating procedure illustrated in the last paragraph of Section 4.3. The number of CB cuts in B+C+V is less than that in B+C, because that valid inequalities help avoid the possibility of infeasible subproblem.

The major issues affecting computation time (or convergence) of Benders algorithm are (1) solving the Benders Master problem (MP), and (2) the quality of Benders cuts. The combinatorial Benders cuts strategy can obtain effective cuts through easier ways, while the valid inequalities can also restrict the solution space of Benders master problem, helping the Benders approach convergent quickly within fewer iterations. The variable reduction tests in BA can significantly reduce the number of Boolean variables before the standard Benders approach starts in step 8 (e.g. on average 65.80% of variables $x_{irs}$), and therefore can significantly save computation time spent on Benders master problems. With all the strategies, the proposed procedure BA has an outstanding performance in terms of both the runtime and the optimality gap, and could satisfy the runtime requirement for practical application in steel corporations.

### 5.2 Results of random data

There is not much difference in the scale of the problems faced at different time by the same corporation because of the heavy and steady demands for bulk materials. Therefore, in order to test the efficiency of the proposed Benders approach for different problem scales, we have generated more test instances based on the real-world data, but of different scales. We consider three levels for the number of materials (10, 15 and 20), two levels for the total number of slots (400 and 600), and two levels for the number of stock rows (5 and 10). The total quantity of all materials to be unloaded in the planning horizon is set to the same level for the same number of materials. Considering different levels of these factors, we have 7 problem groups each with a different combination of factor levels as shown in Table 3. For each problem group 10 instances are generated randomly. For each instance, the number of slots in a row is generated from a range, making sure that the sum of the slots for all the rows equal to the set total. The lengths of the potential fields, the lengths of empty fields and the lengths of the fields occupied by the materials to be retrieved in the planning horizon and their available times are all generated in certain ranges reflecting the situation in practice. The quantity of each material is generated with a practical range, so that the space utilization will be at the set level. The stopping criterion of a 1% optimality gap is also used for these instances.

For each problem group, we summarize the average CPU time and optimality gap in Table 3. The results show that the Benders approach (BA) we proposed outperforms the CPLEX in terms of the runtime. In both methods, the number of materials has a significant influence on the CPU time. Also, for the problem groups that have the same number of materials, both methods are more efficient when the levels of space utilization are higher. The Benders-based approach (BA) we proposed is able to solve problems of large scales in reasonable computational time, and is more efficient for the problems at the same level of space utilization while $|R|$ is larger.

## 6 Conclusion

In this paper, a mixed integer linear programming model is formulated for the integrated storage space allocation and ship scheduling problem in bulk cargo terminal, and a solution procedure based on Benders decomposition algorithm is applied to solve the problem. Two types of valid inequalities are added to the Benders master problem to restrict its solution space and improve the lower bounds efficiently. The combinatorial Benders cut is generated to help break infeasibility and further accelerate convergence. Several variable reduction tests are conducted to identify the variables that would not appear in the optimal solution of MILP, so as to reduce the problem scale. Also, we construct a solution procedure that could get full utilization of the accelerating strategies mentioned above. Experimental results demonstrate that the Benders method we proposed is effective in convergence performance and solution time, and could well respond to the challenges of realistic operations.

## Acknowledgement

This research is partly supported by the Fund for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 71321001), and the State Key Program of the National Natural Science Foundation of China (Grant No. 71032004).

## References

Ago, M., Nishi, T. and Konishi, M. (2007) Simultaneous optimization of storage allocation and routing problems for belt-conveyor transportation, *Journal of Advanced Mechanical Design Systems and Manufaturing*, **1**, 250-261.

Benders, J. F. (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, **4**(1), 238-252.

Bruzzone, A. and Signorile, R. (1998) Simulation and genetic algorithms for ship planning and shipyard layout. *Simulation*, **71**, 74-83.

Cao, J.X., Lee, D.-H., Chen, J.H. and Shi, Q. (2010) The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods. *Transportation Research Part E*, **46**, 344-353.

Codato, G., & Fischetti, M. (2006) Combinatorial Benders' cuts for mixed-integer linear programing. *Oprerations Research*, **54**(4), 756-766.

Contreras, I., Cordeau, J.-F. and Laporte, G. (2011) Benders decomposition for large-Scale uncapacitated hub location. *Oprerations Research*, **59**(6), 1477-1490

Hooker, J.N. (2000) *Logic-Based Methods for Optimization: Combining Optimization and Constraints Satisfaction*, John Wiley and Sons, Singapore, 389-419.

Kim, K.H. and Hong, G.P. (2006) A heuristic rule for relocating blocks. *Computers & Operations Research*, **33**, 940-954.

Kim, B.I., Koo, J. and Park, B.S.(2009) A raw material storage yard allocation problem for a large-scale steelworks. *International Journal of Advanced Manufaturing Technology*, **41**, 880-884.

Kim, B.I., Koo, J. and Sambhajirao, H.P. (2011) A simplified steel plate stacking problem. *International Journal of Production Research*, **49**(17), 5133-5151.

Kim, K.H., Park, Y.M. and Ryu, K.R. (2000) Deriving decision rules to locate export containers in container yards. *European Journal of Operations Research*, **124**, 89-101.

Magnanti, T.L., and Wong, R.T. (1981). Accelerating Benders decomposition: algorithmic enhancement and model selection criteria. *Operations Research*, **29**(3), 464-484.

McDaniel, D. and Devine, M. (1977) A modified Benders partitioning algorithm for mixed integer programming. *Management Science*, **24**(3), 312-319.

Saharidis, G.K.D., Boile, M. and Theofanis, S. (2011). Initialization of the Benders master problem using valid inequalities applied to fixed-charge network problems. *Expert Systems with Applications*, **38**(6), 6627-6636.

Tang, L.X., Jiang, W., Liu, J.Y. and Dong, Y. (2015) Research into container reshuffling and stacking problems in container terminal yards. *IIE Transactions*, **47**(7), 751-766.

Tang, L.X., Jiang, W., and Saharidis, G.K.D. (2013) An improved Benders decomposition algorithm for the logistics facility location problem with capacity expansions. *Annals of Operations Research*, **210**(1), 165-190.

Tang, L.X., Wang, G.S. and Chen, Z.-L. (2014) Integrated charge batching and casting width selection at Baosteel. *Operations Research*, **62**(4), 772-787.

Wan, Y.-w., Liu, J.Y. and Tsai1, P.-c. (2009) The assignment of storage locations to containers for a container stack. *Naval Research Logistics*, **56**, 699-713.

Zhang, C.Q., Liu, J.Y., Wan, Y.-w., Murty, K.G. and Linn, R.J. (2003) Storage space allocation in container terminals. *Transportation Research Part B*, **37**, 883-903

## Appendix

Table A. Index of optimization programs referred in the paper

| Program name | First apperance | Explanation |
|---|---|---|
| **MILP** | Section 3.2 | the problem fomulation in the paper |
| **MP/ SP/ DSP** | Section 4.1 | Benders master/ sub-/ dual sub- problem formulation |
| **MP**$^k$ | Section 4.3.2 | **MP** in the $k^{\text{th}}$ iteration of the Benders approach |
| **MP**$^k{}_{\text{LP}}$ | Section 4.3.2 | relaxed problem of **MP**$^k$ with integer variables relaxed (except variables ***u***) |
| **MILP**$_{\text{LP}}$ | Section 4.3.2 | relaxation problem of **MILP** with integer variables relaxed (except variables ***u***) |
| **A_MILP**$_{\text{LP}}$ | Section 4.3.2 | auxiliary problem constructed from **MILP**$_{\text{LP}}$ with variables ***u*** fixed |
| **MILP_CB** | Section 4.3.2 | auxiliary problem constructed from **A_MILP**$_{\text{LP}}$ by adding an upper bound constraint, so as to construct combinatorial Benders cut |
| **MP**$^k{}_{\text{LP}}$**(*IR*)** | Section 4.4 | **MP**$^k{}_{\text{LP}}$ with the constraints $\sum_{(i,r)\in IR} z_{ir} \geq 1$ for a subset ***IR*** of $\{(i,r) \mid i\in I;\ r\in R\ \}$, so as to carry out variable reduction tests in **Proposition 6** |

## Biographies

Lixin Tang received a B.Eng. degree in Industrial Automation, an M.Eng. degree in Systems Engineering, and a Ph.D. degree in Control Theory and Applications from Northeastern University, Shenyang, China, in 1988, 1991, 1996, respectively. Currently he is a Cheung Kong Scholars Chair Professor and the director of the Institute of Industrial Engneering and Logistics Optimization at Northeastern University, China. He is a Cheung Kong Scholars Chair Professor, and the director of the Institute of Industrial Engneering and Logistics Optimization at Northeastern University, China. His research interests include plant-wide production and logistics planning of smart industry, production and logistics batching and scheduling, operations analytics and optimization for smart industry, optimal and predictive control for cyber-physics system. His research papers have appeared in academic journals such as *Operations Research*, *IIE Transactions*, *Naval Research Logistics*, *IEEE Transactions on Evolutionary Computation*, *IEEE Transactions on Power Systems*, and *European Journal of Operational Research*. He serves as an Associate Editor of *Annals of Operations Research*, *Journal of Scheduling, International Journal of Production Research*, *Journal of the Operational Research Society*, *IEEE Transactions on Cybernetics*, *IEEE Transactions on Automation Science and Engineering*, and an Area Editor of *Asia-Pacific Journal of Operational Research*.

Defeng Sun is a Ph.D. candidate in the Institute of Industrial Engneering and Logistics Optimization at Northeastern University, China. He received a M.S. in System Engineering at Northeastern University, China(2010). His research interests include integer programming, logistics planning and scheduling, inventory control and supply chain design and optimization.

Jiyin Liu is Professor of Operations Management and Director of the MSc Management Programmes portfolio at Loughborough University School of Business and Economics. He is also a Cheung Kong Scholars Visiting Chair Professor in the Institute of Industrial Engineering and Logistics Optimization at Northeastern University, China. He received a Ph.D. degree in Manufacturing Engineering and Operations Management from the University of Nottingham, Nottingham, U.K., in 1993. He has been working on modelling and optimisation of operations planning and scheduling problems in logistics and production systems. His research papers appeared in leading academic journals such as *Operations Research*, *European Journal of Operational Research*, *IIE Transactions, Naval Research Logistics*, *IEEE Transactions on Evolutional Computation* and *Transportation Research Part B: Methodological*.

Table 1 Experimental result of real-world problem instances

| Instance | | CPLEX | B | B+C | B+C+V | BA |
|---|---|---|---|---|---|---|
| 1 | CPU | 09:50:35 | 12:00:00 | 08:25:06 | 06:12:53 | 00:05:24 |
| | Gap(%) | 0.82 | 23.89 | 0.46 | 0.59 | 0.78 |
| 2 | CPU | 09:37:64 | 12:00:00 | 07:03:44 | 06:51:42 | 00:07:56 |
| | Gap(%) | 0.65 | 18.10 | 0.77 | 0.60 | 0.35 |
| 3 | CPU | 09:21:08 | 12:00:00 | 08:05:31 | 07:13:29 | 00:09:53 |
| | Gap(%) | 1.00 | 22.59 | 0.72 | 0.59 | 0.49 |
| 4 | CPU | 08:17:55 | 12:00:00 | 08:51:46 | 06:59:56 | 00:07:15 |
| | Gap(%) | 0.96 | 30.85 | 0.89 | 0.64 | 0.86 |
| 5 | CPU | 10:49:06 | 12:00:00 | 08:12:36 | 06:25:17 | 00:06:31 |
| | Gap(%) | 1.00 | 26.71 | 0.42 | 0.77 | 0.63 |
| 6 | CPU | 08:50:19 | 12:00:00 | 08:34:41 | 08:28:46 | 00:07:18 |
| | Gap(%) | 1.00 | 17.55 | 0.83 | 0.34 | 0.76 |
| 7 | CPU | 10:11:30 | 12:00:00 | 07:35:29 | 06:15:27 | 00:07:03 |
| | Gap(%) | 0.92 | 34.03 | 0.22 | 0.71 | 0.56 |
| 8 | CPU | 09:39:15 | 12:00:00 | 07:27:50 | 06:49:33 | 00:08:21 |
| | Gap(%) | 0.89 | 16.28 | 0.72 | 0.75 | 0.58 |
| 9 | CPU | 08:23:46 | 12:00:00 | 07:24:07 | 06:13:01 | 00:06:44 |
| | Gap(%) | 0.94 | 22.86 | 0.31 | 0.62 | 0.36 |
| 10 | CPU | 09:20:38 | 12:00:00 | 06:43:37 | 06:31:38 | 00:09:15 |
| | Gap(%) | 0.75 | 21.49 | 0.65 | 0.32 | 0.47 |

Table 2 Number of CB cuts and first LBs obtained in different approaches

| Approach | | B | B+C | B+C+V | BA |
|---|---|---|---|---|---|
| number of CB Cuts | | - | 11.5 | 3.8 | 4.6 |
| First LB | LB value | 5082 | 9715 | 10521 | 13654 |
| | Relative | - | 91.16 % | 107 % | 168.67 % |

Table 3 Experimental results of random data

| Group | Scale | | | CPLEX | | BA | |
|---|---|---|---|---|---|---|---|
| | \|*I*\| | \|*R*\| | \|*S*\| | *CPU* | *Gap* | CPU | Gap |
| 1 | 10 | 5 | 300 | 03:06:31 | 0.77 | 0:04:26 | 0.56 |
| 2 | 10 | 5 | 500 | 03:51:27 | 0.89 | 0:05:09 | 0.52 |
| 3 | 15 | 5 | 300 | 07:05:54 | 0.81 | 0:07:19 | 0.67 |
| 4 | 15 | 10 | 300 | 07:31:02 | 0.75 | 0:06:53 | 0.68 |
| 5 | 15 | 10 | 500 | 08:44:39 | 0.82 | 0:07:46 | 0.40 |
| 6 | 20 | 10 | 500 | 12:28:06 | 0.93 | 0:10:55 | 0.61 |

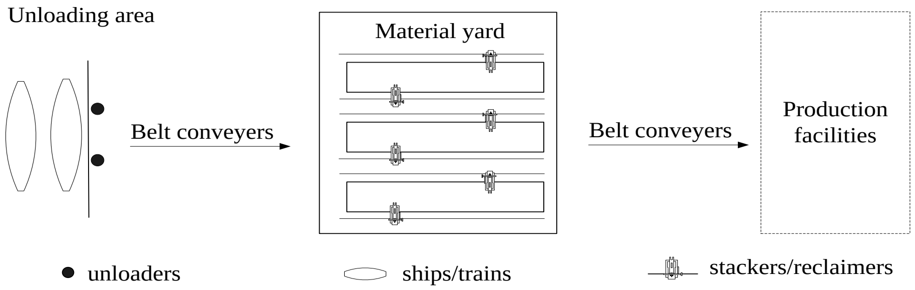


Fig. 1 The schematic flows of bulk materials

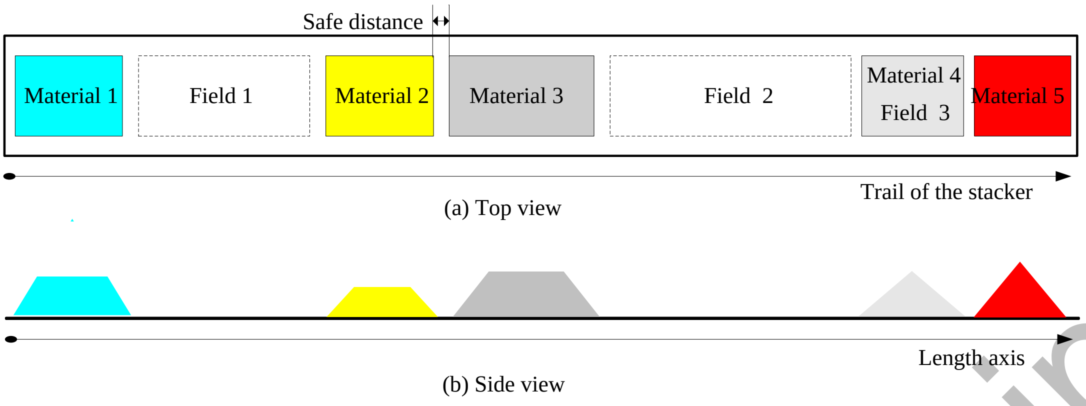
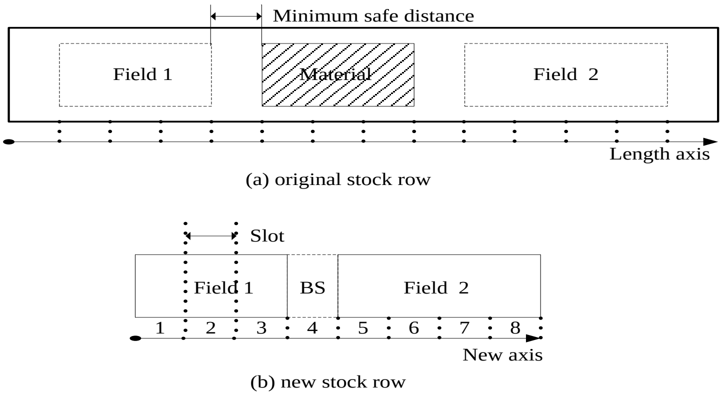


Fig. 2 Top and side views of a stock row



(a) original stock row

(b) new stock row

Fig. 3 Illustration of modeling available fields in a row

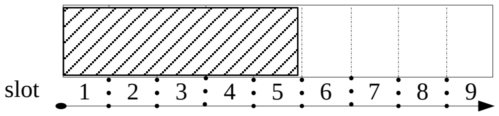


Fig. 4 Allocation structure