World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:6, No:2, 2012

Mathematical Model and Solution Algorithm for Containership Operation/Maintenance Scheduling

Hun Go, Ji-Su Kim and Dong-Ho Lee

***Abstract*—**This study considers the problem of determining operation and maintenance schedules for a containership equipped with components during its sailing according to a pre-determined navigation schedule. The operation schedule, which specifies work time of each component, determines the due-date of each maintenance activity, and the maintenance schedule specifies the actual start time of each maintenance activity. The main constraints are component requirements, workforce availability, working time limitation, and inter-maintenance time. To represent the problem mathematically, a mixed integer programming model is developed. Then, due to the problem complexity, we suggest a heuristic for the objective of minimizing the sum of earliness and tardiness between the due-date and the starting time of each maintenance activity. Computational experiments were done on various test instances and the results are reported.

***Keywords*—**Containerships, operation and preventive maintenance schedules, integer programming, heuristic

I. Introduction

MAINTENANCE is one of functional and operational activities to ensure operational consistency, efficiency and productivity of a system. In general, maintenance is defined as the work performed to keep a system in good condition and working order. The primary goal of maintenance is to avoid or mitigate the consequences of failures and hence provide economical and reliable operation of a system.

There are two basic types of maintenance [1]: corrective maintenance and preventive maintenance. Of the two types, preventive maintenance is the work, including test, measurement, adjustment and part replacement, performed specifically to prevent failures from occurring. It is the activity performed by maintenance workers for the purpose of maintaining a system in a reliable condition by providing systematic inspection, detection, and correction of incipient failures before they occur or they develop into major defects. In general, preventive maintenance policies can be classified into the planned maintenance in which the maintenance is preplanned and the condition-based maintenance in which maintenance is done when one or more indicators show that a system is going to fail or its performance is deteriorating.

Hun Go is with the Department of Industrial Engineering, Hanyang University, Seoul, Korea (e-mail: mucleman2@naver.com).

Ji-Su Kim is with the Department of Industrial Engineering, Hanyang University, Seoul, Korea (e-mail: iekjs@hanyang.ac.kr).

Dong-Ho Lee is with the Department of Industrial Engineering and the Graduate School of Technology and Innovation Management, Hanyang University, Seoul, Korea (corresponding author, phone: 82-2-2220-0475; fax: 82-2-2297-0475; e-mail: leman@ hanyang.ac.kr).

Among various decision problems in preventive maintenance, we focus on the scheduling problem. The problem is to allocate resources over time to perform a set of maintenance activities while considering system-specific requirements such as component operation schedule, available workforce, working time limitation, etc. More specifically, the main decisions are the due-date and the actual start time of each maintenance activity. The due-date of a maintenance activity, determined from the operation schedule of the corresponding component within a system, is the time at which the maintenance activity is to be started. Also, the start time of each maintenance activity, together with its duration, determines the shutdown time of the associated component. Here, the shutdown time, especially important in electric power or chemical plants, is the time over which the associated components are stopped. During the shutdown time, the maintenance activities, such as inspection, repair, test and part replacement, are performed.

The previous studies on preventive maintenance scheduling can be classified into major application areas. For power plants, Satoh and Nara [2] suggest a simulated annealing algorithm that determines the maintenance schedule, i.e., start time of each maintenance activity and generator output for a power generator unit while satisfying the limited workforce and the anticipated power demands for the objective of minimizing the sum of electric production and maintenance costs. Dahal and Chakpitak [3] suggest meta-heuristics that determine the maintenance schedule of a power generator unit under the reliability measure of minimizing the sum of squares of the reserve generation, and Alardhi and Labib [4] develop an integer programming model for generation and desalination units in a cogeneration plant that generates electric power and produces fresh water by desalting sea water at the same time. Transportation facilities area, such as railways, ships and airplanes, are another application area. Higgins [5] report an improvement over the conventional manual method by suggesting a tabu search algorithm for the maintenance and crew scheduling problem in a railway track for the objective of minimizing the disruption to and from scheduled trains and the completion times of maintenance activities. Budai *et al*. [6] suggest heuristic algorithms that determine the schedules of routine (cyclic) and project (non-cyclic) maintenance activities in railway maintenance for the objective of minimizing the sum of track possession and maintenance costs. Also, Joo [7] consider the problem for a modularly designed engine installed in advanced jet trainers with limited spare modules, and suggest a dynamic programming algorithm that minimizes the total opportunity costs associated with premature maintenance. Deris *et al*. [8] consider the problem of determining the schedule of cyclic maintenance activities of a battleship in a squadron for the objective of maximizing the availability of ship operation under time window and resource constraints, and suggest a genetic algorithm after formulating it as a constraint satisfaction problem.

This study considers the preventive maintenance scheduling problem for containerships, called the containership maintenance scheduling problem (C-MSP) in this paper. The problem is to determine operation and maintenance schedules at the same time for components of a containership during its sailing from a start to a destination port. Here, the operation schedule specifies the working times of components and the maintenance schedule specifies the start time of each maintenance activity. The two schedules have a close relationship in that the operation schedule specifies the due-date of each maintenance activity, i.e., the time at which the corresponding maintenance activity is to be started. To cope with the just-in-time maintenance management, this study considers the objective of minimizing the sum of deviations between the due-dates and the start times of maintenance activities, i.e., total earliness and tardiness.

To the best of authors' knowledge, there is no previous study on ship maintenance scheduling while a ship sails according to a navigation schedule, i.e., sequence of ports to be visited. For example, Deris *et al*. [8] consider a ship maintenance scheduling problem while the ship is in a dockyard, i.e., maintenance activities of a ship are carried out only when it is in dockyard. Unlike this, we consider the problem while a ship is on a voyage. Also, compared with other maintenance scheduling for plants and railways, the C-MSP considered in this study has containership-specific considerations, e.g., navigation schedule, component operation schedule, maintenance activity types, working time limitation, workforce availability, inter-maintenance time constraint, etc.

To represent the problem mathematically, a mixed integer programming model is suggested that incorporates the containership-specific constraints. Then, due to the complexity of the problem, we suggest a heuristic that determines operation and maintenance schedules at the same time. The heuristic suggested in this study consists of two main phases: constructing an initial solution and improvement. To show the performance of the heuristic, computational experiments were done on various test instances and the results are reported.

This paper is organized as follows. In the next section, the problem is described in more details and the corresponding mixed integer programming model is suggested. Section 3 presents the heuristic algorithm, and the results on computational tests are reported in Section 4. Finally, Section 5 concludes the paper with a summary and discussion of future research.

II. Problem Description

To clarify the suggested problem, we first explain the characteristics of containerships, i.e., components of a containership and two maintenance types. Then, we explain the problem in more details with a mathematical model.

A containership has parallel and identical components, e.g., four generator engines, two fuel oil purifiers, two lubricating purifiers, three air compressors, etc. During a sailing, some of parallel components are operated according to the operation schedule that specifies their working times. Also, the maintenance activities of a containership, performed during its sailing, can be classified into two basic types: (a) routine maintenance activities; and (b) operation-level maintenance activities. A routine activity, such as simple inspection and repair, is cyclic and deterministic one, while operation-level activity, such as replacement of a part, is the one performed when the cumulative usage time reaches a pre-determined value. For example, an inspection of a cam shaft within the generator engine is a routine maintenance activity while a replacement of a piston ring within the main propulsion engine is an operation-level maintenance activity.

The problem considered here can be briefly described as follows: *for a given planning horizon, the problem is to determine operation and maintenance schedules for components within a containership while satisfying component requirements and containership-specific constraints for the objective of minimizing the total earliness and tardiness*. Here, the earliness (tardiness) for a maintenance activity occurs when its start time is less (greater) than its due-date. In general, the concept of earliness and tardiness emerges from the just-in time production and hence the objective pursues timely maintenance. In the practical aspect, the earliness reduces the level of worker satisfaction and the tardiness increases the probability of component breakdowns.

It is assumed that the planning horizon, determined by the navigation schedule, consists of discrete periods, e.g., 1 month (planning horizon) with 30 days (periods). For each maintenance activity, the due-date, i.e., the period at which the activity is to be started, is determined by the component operation schedule. Also, the start time, i.e., the period at which the maintenance activity will be actually started, is determined by the maintenance schedule. Therefore, the two decision variables, operation and maintenance schedules, are closely interrelated. Recall that the due-date of each operation-level maintenance activity of a component is determined by the number of periods for which the component is to be operated, i.e., cumulative usage time.

The C-MSP considered in this study has four constraints: (a) component requirements; (b) workforce availability; (c) working time limitation; and (d) inter-maintenance time constraint. First, the component requirements for each type in a period imply the number of components that must be operated in that period. In general, the component requirements are obtained from the basic load condition that specifies the smallest component requirements for normal operation of a containership. Therefore, a containership cannot be operated properly unless this constraint is satisfied. For example, one of parallel components must be operated during sailing and all components must be operated for safety when a containership enters or leaves a port. Second, the workforce availability implies the restriction on the number of available workers in each period. Note that a maintenance activity may be done by one or more workers. Third, the working time limit implies the upper limit on the working time in each period. Finally, the inter-maintenance time constraint implies that the interval between two consecutive activities must be large than or equal to a pre-specified number of periods.

It is assumed that there is no change in the sailing schedule of a containership. Also, the duration of each maintenance activity is deterministic and given in advance. Besides these, other assumptions made are summarized as follows: (a) each worker can perform at most one maintenance activity at a time; (b) maintenance activities are non-preemptive, i.e., once an activity is started, it will stay without interruption until its completion; (c) usage times of components are deterministic and given in advance; and (d) inter-maintenance time of each maintenance activity is deterministic and given in advance.

To represent the problem more clearly, a mixed integer programming model is suggested in this study. Before presenting the model, the notations used are summarized below.

Indices

m component types, m = 1, 2, 3, … M
i components, i = 1, 2, 3, … I_m
a maintenance activities, $a \in A_i$, where $A_i = A_i' \cup A_i''$ (A_i' and A_i'' denote the sets of routine and operation-level maintenance activities, respectively.)
d period (day unit), d = 1, 2, … D
k period (hour unit), k = 1, 2, 3, … K, where $K = D \cdot R$ (R denotes the number of working hours per day.)
q frequency of maintenance activity, q = 1, 2, 3, … FR_{mia}, where FR_{mia} denotes the number of activities of type a on component i of type m during the planning horizon

Parameters

OP_{mk} requirement of component type m in period k
TO_{mi} total usage time of component i of type m over the planning horizon, which can be obtained from component requirement OP_{mk}
DU_{mia} duration of maintenance activity a on component i of type m
C_{mia} number of inter-maintenance periods for maintenance activity a on component i of type m ($C_{mia} \geq V_{mia}$, where V_{mia} denotes the minimum number of inter-maintenance periods for maintenance activity a on component i of type m)
G_{mia} number of elapsed periods for maintenance activity a on component i of type m at the beginning of the planning horizon
WF_{mia} number of workers needed to perform activity a on component i of type m
WT working hour in a day
WF available workforce

Decision variables

X_{mik} = 1 if component i of type m operates in period k, and 0 otherwise
DD_{miaqk} = 1 if due-date of qth maintenance activity a on component i of type m is fixed in period k, and 0 otherwise
S_{miaqk} = 1 if qth maintenance activity a on component i of type m starts in period k, and 0 otherwise
Y_{miaqk} = 1 if qth maintenance activity a on component i of type m is performed in period k, and 0 otherwise

Now, the mixed integer programming model is given below.

$$[\mathbf{P}]\ \text{Minimize} \sum_{m=1}^{M}\sum_{i=1}^{I_m}\sum_{a\in A}\sum_{q=1}^{FR_{mia}}(E_{miaq}+T_{miaq})$$

subject to

$$\sum_{i=1}^{I_m} X_{mik} = OP_{mk} \qquad \text{for all } m \text{ and } k \tag{1}$$

$$\sum_{k=1}^{K} X_{mik} = TO_{mi} \qquad \text{for all } m \text{ and } i \tag{2}$$

$$\sum_{k=1}^{k'} X_{mik} \geq (C_{mia}\cdot q - G_{mia})\cdot DD_{miaqk'} \qquad \text{for all } m, i, a \in A'', q, \text{ and } k' \tag{3}$$

$$\sum_{k=1}^{k'} X_{mik} \leq TO_{mi} - [TO_{mi} - (C_{mia}\cdot q - G_{mia})]\cdot DD_{miaqk'} \qquad \text{for all } m, i, a \in A'', q, \text{ and } k' \tag{4}$$

$$X_{mik} \geq DD_{miaqk} \qquad \text{for all } m, i, a \in A'', q, \text{ and } k \tag{5}$$

$$(C_{mia}\cdot q - G_{mia})\cdot DD_{miaqk'} \leq k' \qquad \text{for all } m, i, a \in A', q, \text{ and } k' \tag{6}$$

$$[K - (C_{mia}\cdot q - G_{mia})]\cdot DD_{miaqk'} \leq K - k' \qquad \text{for all } m, i, a \in A', q, \text{ and } k' \tag{7}$$

$$\sum_{k=1}^{K} DD_{miaqk} = 1 \qquad \text{for all } m, i, a \text{ and } q \tag{8}$$

$$\sum_{k=1}^{K} S_{miaqk} = 1 \qquad \text{for all } m, i, a \text{ and } q \tag{9}$$

$$\sum_{k=1}^{K} k\cdot S_{miaqk} + E_{miaq} - T_{miaq} = \sum_{k=1}^{K} k\cdot DD_{miaqk} \qquad \text{for all } m, i, a \text{ and } q \tag{10}$$

$$\sum_{k=1}^{K}(k+V_{mia})\cdot S_{miaqk} \leq \sum_{k=1}^{K} k\cdot S_{mia(q+1)k} \qquad \text{for all } m, i, a \text{ and } q \tag{11}$$

$$X_{mik} + Y_{miaqk} \leq 1 \qquad \text{for all } m, i, a, q \text{ and } k \tag{12}$$

$$X_{mik} + S_{miaqk} \leq 1 \qquad \text{for all } m, i, a, q \text{ and } k \tag{13}$$

$$1 - Y_{miaqk'} = \begin{cases} 1 - k' + (d-1)\cdot R + \displaystyle\sum_{k=(d-1)\cdot R+1}^{k'}(1 - S_{miaqk}) & (d-1)\cdot R + 1 \leq k' \leq (d-1)\cdot R + DU_{mia} \\ 1 - DU_{mia} + \displaystyle\sum_{k=1+k'-DU_{mia}}^{k'}(1 - S_{miaqk}) & (d-1)\cdot R + DU_{mia} + 1 \leq k' \leq d\cdot R \end{cases}$$
$$\text{for all } m, i, a, q \text{ and } d \tag{14}$$

$$\sum_{k=(d-1)\cdot R+1}^{d\cdot R - DU_{mia}+1} S_{miaqk} \leq 1 \qquad \text{for all } m, i, a, q \text{ and } d \tag{15}$$

$$\sum_{k=d\cdot R - DU_{mia}+2}^{d\cdot R} S_{miaqk} = 0 \qquad \text{for all } m, i, a \text{ and } q \tag{16}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{I_m}\sum_{a\in A}\sum_{q=1}^{FR_{mia}} WF_{mia}\cdot Y_{miaqk} \le WF \qquad \text{for all } k \tag{17}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{I_m}\sum_{a\in A}\sum_{q=1}^{FR_{mia}}\sum_{k=(d-1)\cdot R+1}^{d\cdot R} WF_{mia}\cdot Y_{miaqk} \le WT \qquad \text{for all } d \tag{18}$$

$$X_{mik} \in \{0,1\} \qquad \text{for all } m, i \text{ and } k \tag{19}$$

$$Y_{miaqk}, S_{miaqk}, DD_{miaqk} \in \{0,1\} \quad \text{for all } m, i, a, q \text{ and } k \tag{20}$$

$$E_{miaq}, T_{miaq} \ge 0 \qquad \text{for all } m, i, a \text{ and } q \tag{21}$$

The objective function, together with constraint (10), minimizes the total earliness and tardiness. Constraints (1) and (2) ensure that both component requirements and total usage times be satisfied. Constraints (3), (4) and (5) specify the due-date of each operation-level maintenance activity, i.e., $a \in A''$. More specifically, constraints (3) and (4) specify the possible periods that the due-date is set from the first to the last period and from the last to the first period, respectively. Also, we can specify the specific due-date with constraints (5) and (8). Here, constraint (8) ensures that only one due-date is assigned to each maintenance activity. Similarly, constraints (6) and (7) specify the due-date of each routine activity. Constraint (9) ensures that one start period is assigned to each maintenance activity. Constraint (10) specifies earliness and tardiness of each maintenance activity, and constraint (11) ensures that the inter-maintenance periods between two consecutive activities be satisfied. Constraints (12) and (13) ensure that no component works when the corresponding maintenance activity is done. Constraint (14) ensures that a maintenance activity is done by the amount of its duration without interruption on daily basis. Also, constraints (15) and (16) specify the range of period in which each maintenance activity can be started. Constraints (17) and (18) represent the workforce and the working time limitations, respectively. Finally, the remaining constraints represent the conditions of the decision variables.

The optimal solutions can be obtained by solving the model [P] using a commercial software package. However, it is not practical due to excessive computation time. In fact, we can easily see that the problem [P] is NP-hard since it contains the well-known knapsack constraints, i.e., the workforce and the working time constraints.

III. Solution Algorithm

The solution algorithm suggested in this study consists of two phases: (a) constructing an initial solution; and (b) improvement. In the construction phase, the due-date of each maintenance activity is assigned to a certain period after fixing the component operation schedule and then the start time of each maintenance activity is determined as the nearest period from its due-date while considering the relevant constraints. In the improvement phase, the initial solution is improved by changing due-dates of maintenance activities, i.e., changing component operation schedules.

A. Constructing an initial solution

In this phase, an initial solution is obtained by three main steps: (a) generating the component operation schedule; (b) assigning the due-date of each maintenance activity; and (c) determining the start time of each maintenance activity.

The component operation schedule (X_{mik}) is generated according to component requirements OP_{mk} and total usage time TO_{mi}. Two cases, single and parallel components for each component type, are considered in this step. In the case of single component, its operation schedule is fixed to its component requirements, i.e., $X_{mik} = OP_{mk}$. On the other hand, in the case of parallel components, the operation schedules are generated randomly while satisfying their component requirements and total usage time, i.e., constraints (1) and (2).

The due-date of each maintenance activity (DD_{miaqk}) is assigned differently according to the maintenance types. In the case of operation-level activities, their due-dates are determined according to cumulative usage times. More specifically, they are determined among those obtained by constraints (3), (4) and (5) that specify the possible periods to which the due-dates of operation-level activities can be assigned. Similarly, the due-dates of routine activities are determined using constraints (6) and (7) are satisfied.

The start time of each maintenance activity (S_{miaqk}) is determined as follows. First, the maintenance activities are sorted using a priority rule. Here, the activities are sorted in the non-decreasing order of frequency index q in the identical maintenance activity. For this purpose, the following three priority rules are suggested in this study.

DU	higher priority given to the activity with larger duration
WF	higher priority given to the activity with larger number of required workers
DW	higher priority given to the activity with larger one among those obtained by multiplying duration and number of workers

Second, the start time of each maintenance activity is determined according to the sorted list. This is done differently according to component types: single and parallel. Note that the start times of the activities associated with single components are determined first, and then those associated with parallel components are determined. The detailed method is explained below.

In the case of single component, the start time is set to the period that gives the smallest earliness or tardiness while satisfying the relevant constraints, i.e.,

$$\underset{k' \in SP_{miaq}}{\arg\min} \left| \sum_{k=1}^{K} k \cdot DD_{miaqk} - k' \right|, \tag{22}$$

where SP_{miaq} is the set of periods that the qth maintenance activity a on component i of type m can be started while satisfying the constraints (11), (14), (15), (16), (17), and (18). If $SP_{miaq} = \varnothing$, the start time of the directly preceding ($q-1$)th activity is set to the period that gives the second smallest earliness or tardiness and then the start time of the qth activity is set. This is done repeatedly until feasible start times are determined. Ties are broken arbitrarily. As in the single components, the start time of each parallel component is set to the period that gives the smallest earliness or tardiness. However, the operation schedule of the parallel components must be changed when the start time does not satisfy the constraints (12), (13) and (14), i.e., no component works when the corresponding maintenance activity is done and a maintenance activity is done by the amount of its duration without interruption. In other words, if a component operates during the maintenance activity, i.e., $X_{mik} = 1$ for some i and k such that $S_{miaqk} \le k \le S_{miaqk} + DU_{mia} - 1$, the operation schedule of the corresponding parallel components is changed until the start time satisfies the constraints. More formally, a new component i' ($\neq i$) of the same type is selected that satisfy the following condition (from the smallest to the largest indexed one)

$$X_{mi'k} = 0 \text{ and } \sum_{a\in A} \sum_{q=1}^{FR_{mia}} Y_{mi'aqk} = 0 \tag{C1}$$

which ensure that no operation and maintenance activities be done on the new component i' in the current period k. Then, for the selected component i', the new period k' is selected (from the smallest to the largest indexed one) in such a way that the following conditions hold.

$$X_{mik'} = 0 \text{ and } \sum_{a\in A} \sum_{q=1}^{FR_{mia}} Y_{miaqk'} = 0 \tag{C2}$$

$$X_{mi'k'} = 1 \tag{C3}$$

$$S_{miaqk} + DU_{mia} \le k' \le K\ , \tag{C4}$$

Here, condition (C2) ensures that no operation and maintenance activities are done on component i' in period k', and conditions (C3) and (C4) specify the set of possible new periods. If the set of possible periods is empty, another new period k' is selected (from the smallest to the largest indexed one) that satisfies (C1), (C2), (C3) and $1 \le k' \le S_{miaqk} - 1$ for the selected component i'. For the current and the selected components i and i' in the current and the new periods k and k', their operation schedules are changed as follows.

$$X_{mik} = 0,\ X_{mi'k} = 1,\ X_{mik'} = 1 \text{ and } X_{mi'k'} = 0$$

Then, the start time of the current activity is determined using the method explained earlier. Here, the due-dates of the operation-level activities for the other components of the same type are re-assigned those specified by constraints (3), (4) and (5) since their operation schedules have been changed.

Now, the detailed procedure for the construction phase is given below.

Phase I. (Constructing an initial solution)

Step 1. (Generate the component operation schedule)
For each component type, do:
- (a) If it is a single component, set its operation schedule to its component requirements. Otherwise, go to Step 1(b).
- (b) Generate the operation schedule of parallel components randomly so as to satisfy their component requirements and total usage times, i.e., constraints (1) and (2).

Step 2. (Assign the due-date of each maintenance activity)
For each maintenance activity, do:
- (a) If the current activity is operation-level, assign its due-date by those specified by constraints (3), (4) and (5). Otherwise go to Step 2(b).
- (b) Assign its due-date by those specified by constraints (6) and (7).

Step 3. (Determine the start time of each maintenance activity)

Step 3.1 Sort maintenance activities with a priority rule. Ties are broken with the increasing order of frequency index.

Step 3.2 From the first to the last activity in the sorted list, do:
- (a) If the activity is on a single component, do:
  - (a-1) Specify the set SP_{miaq} of periods that the activity can be started while satisfying the constraints (11), (14), (15), (16), (17) and (18) except for the perio ds considered previously.
  - (a-2) If $SP_{miaq} \neq \emptyset$, set its start time to the period that gives the smallest earliness or tardiness from its due-date, i.e., condition (22). Otherwise, set $q = q - 1$ and go to Step 3.2(a-1).
- (b) Otherwise (parallel components), do:
  - (b-1) Specify the set SP_{miaq} of periods that the activity can be started while satisfying the constraints (11), (15), (16), (17) and (18) except for the periods considered previously.
  - (b-2) If $SP_{miaq} \neq \emptyset$, set its start time to the period that gives the smallest earliness or tardiness from its due-date. Otherwise, set $q = q - 1$ and go to Step 3.2(b-1).
  - (b-3) If the start time of the current activity does not satisfy constraints (12), (13) and (14), i.e., the component operates during the maintenance activity, select a new component of the same type and the new period and change the operation schedules of the current and the selected components in the current and the new periods using the method explained earlier. Then, set its start time as in Step 3.2(b-2). After reassigning the due-dates of the operation-level activities for the other components of the same type randomly among those specified by constraints (3), (4) and (5), go to Step 3.2(b-1).

B. Improvement

In this phase, the initial solution is improved by changing the component operation schedule of operation-level maintenance activities on parallel components while their start times and due-date are remained unchanged.

First, the operation-level maintenance activities are sorted using the method explained in Step 3.1 of Phase I, i.e., sort maintenance activities using a priority rule (DU, WF or DW), with breaking ties in the increasing order of frequency index.

Second, according to the sorted list, the operation schedules of the parallel components associated with the current activity are changed in such a way that the deviation between the start time and the due-date is reduced. Note that the start times (S_{miaqk}) fixed in phase 1 are not changed. Two cases, tardy and early maintenance activities, are considered in this step. If the current activity is tardy, i.e., $S_{miaqk} > DD_{miaqk}$, the operation schedules of the component associated with the current activity and another component of the same type are changed if it gives an improved solution. More formally, the current period k is selected (from the smallest to the largest indexed one) such that

$$X_{mik} = 1 \text{ and } 1 \le k \le DD_{miaqk} \quad \text{(C5)}$$

i.e., component i operates in the current period k before the due-date of the current activity. Then, a new component i' ($\neq i$) of the same type is selected (from the smallest to the largest indexed one) that satisfy the following condition

$$X_{mi'k} = 0 \text{ and } \sum_{a \in A} \sum_{q=1}^{FR_{mia}} Y_{mi'aqk} = 0 \,, \quad \text{(C6)}$$

which means that no operation and maintenance activities are done on the new component i' in the current period k and the new period k' is selected (from the smallest to the largest indexed one) that satisfy the following conditions.

$$X_{mik'} = 0 \text{ and } \sum_{a \in A} \sum_{q=1}^{FR_{mia}} Y_{miaqk'} = 0 \quad \text{(C7)}$$

$$X_{mi'k'} = 1 \text{ and } DD_{miaqk} \le k' \le K \quad \text{(C8)}$$

Here, conditions (C7) and (C8) imply that no operation and maintenance activities are done on the current component i in the new period k' and component i' must operate in the new period k' after the due-date of the current activity, respectively. Finally, the current solution is updated if we can obtain an improved solution after changing the operation schedules of components i and i' in periods k and k' as follows.

$$X_{mik} = 0,\ X_{mi'k} = 1,\ X_{mik'} = 1 \text{ and } X_{mi'k'} = 0\,,$$

On the other hand, if the current activity is early, i.e., $S_{miaq} < DD_{miaq}$, selected are a new component i' ($\neq i$), together with periods k and k' that satisfy (C6), (C7) and

$$X_{mik} = 1 \text{ and } DD_{miaqk} \le k \le K \quad \text{(C9)}$$

$$X_{mi'k'} = 1 \text{ and } 1 \le k' \le DD_{miaqk} \quad \text{(C10)}$$

The others are the same as the tardy case.

Now, the detailed procedure for the improvement phase is summarized below.

Phase II. (Improvement)

Step 1. Sort the operation-level maintenance activities as in Step 3.1 of Phase I.

Step 2. From the first to the last activity in the sorted list, do:

(a) If the current activity is tardy, i.e., $S_{miaq} > DD_{miaq}$, select a new component i' ($\neq i$), together with periods k and k', which satisfy the conditions (C5), (C6), (C7) and (C8). Otherwise, select a new component i' ($\neq i$), together with periods k and k', which satisfy the conditions (C9), (C6), (C7) and (C10).

(b) If an improved solution is obtained after changing the operation schedules of components i and i' in periods k and k' as

$$X_{mik} = 0,\ X_{mi'k} = 1,\ X_{mik'} = 1 \text{ and } X_{mi'k'} = 0$$

update the current solution.

IV. Computational Experiments

To show the performance of the heuristic suggested in this study, computational experiments were carried out and the results are reported in this section. Note that we tested three heuristics according to the three priority rules that sort the maintenance activities. The heuristics were coded in C, and tests were carried out on a personal computer with Intel i5 processor operating at 2.80 GHz clock speed.

For the test, we generated 80 small sized instances, i.e., 10 instances for each of eight combinations of two levels of workforce capacity (tight with 8 workers and loose with 10 workers), two levels of the number of periods (24: 3 days with 8 hours/day and 30: 3 days with 10 hours/day) and two levels of component types (3 and 4). The number of identical components for each component type was generated from $DU(1, 3)$, where $DU(a, b)$ denotes the discrete uniform distribution with range $[a, b]$. The component requirements in each period of the planning horizon (OP_{mk}) were generated randomly using the basic load condition. Also, the number of maintenance activities for each component (i) was generate $DU(1, 4)$. For instances with 24 periods, the duration (DU_{mia}), the inter-maintenance time (C_{mia}), the number of required workers (WF_{mia}), the elapsed number of periods (G_{mia}) and the minimum inter-maintenance time (V_{mia}) of each maintenance activity were generated from $DU(1, 3)$, $DU(7, 16)$, $DU(1, 3)$, $DU(0, 9)$ and $DU(0, 1)$, respectively. Similarly, for the instances with 30 periods, they were generated from $DU(1, 4)$, $DU(11, 22)$, $DU(1, 4)$, $DU(0, 10)$ and $DU(1, 2)$, respectively.

Test results for the small sized instances are summarized in Table 1 that shows the percentage gaps from the optimal solution values (or lower bounds) and CPU seconds. Here, the optimal solutions (or lower bounds) were obtained by solving the mixed integer programming model [P] using CPLEX 11.0 in 3600 seconds. It can be seen from the table that among the three rules that sort the maintenance activities, the DW rule was slightly better than the others in overall average, especially when the workforce capacity is loose. However, no one rule dominates the others. In fact, its overall average gaps were 10.98% and 8.08% for the instances with tight and loose workforce capacities, respectively. Also, the gaps get higher as the number of component types increases. Finally, compared with CPLEX 11.0, the CPU seconds of the heuristics were very short.

Also, we tested the heuristics on 30 additional medium to large sized instances, i.e., 10 instances for each of three combinations of the numbers of components types and periods (5 component types with 96 periods, 6 component types with 120 periods, and 7 component types with 144 periods). The workforce capacity was set to 8 workers. The number of activities for each component, the duration, the elapsed periods and the number of required workers were generated from $DU(1, 3)$, $DU(1, 6)$, $DU(2, 6)$, $DU(0, 28)$ and $DU(1, 5)$, respectively. Also, the inter-maintenance times and the minimum inter-maintenance times for the instances with 96 (120, 144) periods were generated from $DU(38, 95)$ ($DU(48, 109)$, $DU(85, 153)$) and $DU(3, 9)$ ($DU(4, 10)$, $DU(8, 15)$), respectively. Since we could not obtain the optimal solutions or effective lower bounds for medium to large sized instances, the three heuristics were compared using the relative performance ratio. Here, the relative performance ratio of heuristic a for an instance is defined as

$$100\cdot(C_a - C_{best}) / C_{best},$$

where C_a is the objective value obtained using the heuristic and C_{best} is the best solution value among those obtained from the three heuristics.

Results of the medium to large sized instances are summarized in Table II that shows the average relative performance ratios of the three heuristics. As in those for the small sized instances, no one rule dominates the others. However, among the three priority rules, the WF rule (that gives higher priority given to the activity with larger number of required workers) was better than the others. This is because the test instances are tight in the number of required workers. Therefore, the results cannot be generalized.

TABLE I
TEST RESULTS ON SMALL SIZED INSTANCES

Number of periods	Number of component types	Workforce capacity											
		Tight (8 workers)						Loose (10 workers)					
		Priority rules						Priority rules					
		DU[1]		WF[2]		DW[3]		DU		WF		DW	
		Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU
24 (3 days, 8 hours)	3	7.4 (0.0, 20.0)*	1.7**	5.9 (0.0, 13.3)	1.7	7.2 (0.0, 18.8)	1.7	6.6 (0.0, 32.0)	1.8	4.8 (0.0, 12.0)	2.2	5.2 (0.0, 12.0)	1.8
	4	15.1 (2.4, 32.1)	2.9	17.4 (2.4, 40.0)	3.0	14.2 (2.4, 32.1)	3.1	11.6 (3.7, 22.6)	3.3	10.0 (3.7, 20.0)	3.4	9.5 (3.7, 20.0)	3.3
30 (3 days, 10 hours)	3	3.4 (0.0, 17.4)	1.9	5.9 (0.0, 13.3)	1.9	7.6 (0.0, 20.0)	1.8	5.1 (0.0, 20.0)	1.4	8.0 (0.0, 26.7)	1.4	6.8 (0.0, 26.7)	1.4
	4	18.7 (0.0, 45.5)	3.2	15.0 (0.0, 29.6)	3.5	14.9 (0.0, 29.6)	3.2	13.0 (0.0, 30.4)	2.5	11.4 (0.0, 37.5)	2.7	10.8 (0.0, 30.4)	2.4
Average		11.2	2.4	11.1	2.5	11.0	2.4	10.7	2.3	8.5	2.4	8.1	2.2

* average gap (min, max) from the optimal solution values or lower bounds out of 10 instances
** average CPU second out of 10 instances
[1] DU: sort the maintenance activities in the non-increasing order of the duration, i.e., DU_{mia}
[2] WF: sort the maintenance activities in the non-increasing order of the number of required workers, i.e., WF_{mia}
[3] DW: sort the maintenance activities in the non-increasing order of the value obtained by multiplying duration and number of workers, i.e., $DU_{mia}\cdot WF_{mia}$

TABLE II
TEST RESULTS ON MEDIUM TO LARGE SIZED INSTANCES

Instance size (M^a, K^b)	Priority rule					
	DU		WF		DW	
	RPR	CPU	RPR	CPU	RPR	CPU
(5, 96)	5.6 (0.0, 28.3)*	20.5	0.7 (0.0, 4.8)	19.7	5.1 (0.0, 20.0)	19.6
(6, 120)	8.0 (0.0, 23.9)	110.6	3.4 (0.0, 22.4)	110.7	13.6 (0.0, 36.4)	112.3
(7, 144)	1.7 (0.0, 6.8)	166.6	2.9 (0.0, 9.7)	170.6	2.0 (0.0, 6.9)	165.3
Average	5.2	99.2	2.3	100.3	6.7	99.1

[a] number of component types
[b] number of periods
* average relative performance ratio (min, max) out of 10 instances
See the footnotes of TABLE I.

V. Concluding Remarks

This study considered the problem of determining component operation as well as maintenance schedules for a containership during its sailing from start to destination port. The two schedules have a close relationship in that the operation schedule specifies the due-date of each maintenance activity, i.e., the time at which the corresponding maintenance activity is to be started. Also, the maintenance schedule specifies the start time of each maintenance activity. To perform timely maintenances, we consider the objective of minimizing the sum of deviations between the due-dates and the start times of maintenance activities, i.e., total earliness and tardiness. Compared with other maintenance scheduling for plants and railways, the problem considered in this study has containership-specific considerations such as sailing schedule, component operation schedule (due-date of each maintenance activity), maintenance activity types (routine and operation-level), workforce availability, working time limitation, inter-maintenance time constraint. To represent the problem mathematically, a mixed integer programming model is suggested. Then, due to the complexity of the problem, we suggest a heuristic algorithm together with three priority rules. Computational experiments were done on a number of randomly generated test instances and the results showed that the heuristic can give reasonable quality solutions.

This research can be extended in several directions. First, one may consider other features, e.g., bunkering of containership, general load of components, etc. Second, it is needed to develop more sophisticated algorithms, especially the method to generate more efficient component operation schedules. Finally, it is needed to check the potential applications of the model and the algorithm to other systems such as plants, other transportation equipment, etc.

References

[1] Wang, H., "A survey of maintenance policies of deteriorating systems," *European Journal of Operational Research*, vol.139(3), pp.469–489, 2002.

[2] Satoh, T., Nara, K., "Maintenance scheduling by using simulated annealing method," *IEEE Transactions on Power Systems*, vol.6(2), pp.850-857, 1991.

[3] Dahal, K.P., Chakpitak, N., "Generator maintenance scheduling in power systems using metaheuristic-based hybrid approaches," *Electric Power Systems Research*, vol.77(7), pp.771–779, 2007.

[4] Alardhi, M., Hannam, R.G., Labib, A.W., "Preventive maintenance scheduling for multi-cogeneration plants with production constraints," *Journal of Quality in Maintenance Engineering*, vol.13(3), pp.276-292, 2007.

[5] Higgins, A., "Scheduling of railway track maintenance activities and crews," *Journal of the Operational Research Society*, vol.49(10), pp.1026-1033, 1998.

[6] Budai, G., Huisman, D., Dekker, R., "Scheduling preventive railway maintenance activities," *Journal of the Operational Research Society*, vol.57(9), pp.1035-1044, 2006.

[7] Joo, S.J., "Scheduling preventive maintenance for modular designed components: A dynamic approach," *European Journal of Operational Research*, vol.192(2), pp.512–520, 2009.

[8] Deris, S., Omatu, S., Ohta, H., Kutar, C.S., Samat, P.A., "Ship maintenance scheduling by genetic algorithm and constraint-based reasoning," *European Journal of Operational Research*, vol.112(3), pp.489-502, 1999.