This is the Pre-Published Version.

This is a post-peer-review, pre-copyedit version of an article published in Annals of Operations Research. The final authenticated version is available online at: https://doi.org/10.1007/s10479-020-03632-6.

Optimal Bi-criterion Planning of Rescue and Evacuation Operations for Marine Accidents Using an Iterative Scheduling Algorithm*

Chi To Ng[1], T.C.E. Cheng[1], Eugene Levner[2] and Boris Kriheli[3]

[1]Logistics Research Centre, Department of Logistics and Maritime Studies
The Hong Kong Polytechnic University, Hong Kong SAR, China
{daniel.ng; edwin.cheng}@polyu.edu.hk

[2]Department of Computer Science
Holon Institute of Technology, Holon, Israel
levner@hit.ac.il

[3]Department of Economics
Ashkelon Academic College, Ashkelon, Israel
borisk@hit.ac.il

Abstract

We consider the problem of real-life evacuation of people at sea. The primary disaster response goal is to minimize the time to save all the people during the evacuation operation, taking into account different groups at risk (children, women, seniors etc) and the evacuation processing time (including the routing time), subject to a budget constraint. There are different evacuation tools (e.g., lifeboats, salvage ships, sea robots, helicopters etc) for rescuing groups at risk to some safe points (e.g., hospitals, other ships, police offices etc). The evacuation processing time of a group at risk depends on the group and the evacuation tool used. The secondary goal is to minimize the cost among all the alternative optimal solutions for the primary goal. We present a new mathematical rescue-evacuation model and design a fast solution method for real-time emergency response for different population groups and different evacuation tools, based on iterative utilization of a modification of the scheduling algorithm introduced by Leung and Ng (2017).

Key words: bi-criterion planning; disaster management; disaster response; accidents at sea; scheduling algorithm

---

*This paper is a corrected, extended, and enriched version of the paper accepted for presentation at the 353rd International Conference on Management, Economics & Social Science - ICMESS 2019, Phonom Penh, Cambodia, 4-5 February 2019.

I. INTRODUCTION

Research interest in the planning and scheduling of disaster response operations, especially emergency evacuation and saving of people during disasters at sea, has increased dramatically over the past decade [1-6]. This paper discusses how advanced computing tools can be applied to improve the rescue and evacuation operations in the disaster response process at sea.

Over the past decades, there has been a significant growth in the number of passengers cruising on large ocean ships all over the world. According to statistics, about 25.8 million passengers had cruised globally in 2017 and the demand for cruising has increased 20.5 per cent in the last five years [1].

With the growth of cruises, the number of accidents and emergencies on board passenger ships in the open sea has also increased, causing large damages to the environment and losses of human lives. Hence, real-life security problems have become a great challenge both for researchers and practitioners. In the academia, research interests in the planning, scheduling, and control of emergency response operations, especially people evacuation from large ships, have increased substantially.

In order to prevent in the future tragic accidents like those of Titanic happening in April 1912, the International Marine Organization (IMO) has issued the international convention of Safety of Life at Sea (SOLAS), and improved several times relevant safe specifications for the personnel evacuation plan at the Maritime Safety Committee (MSC) [2].

The literature on emergency evacuation-from-ship models is vast. The formal emergency routing models can be generally classified into two categories, namely analytical and simulation models. Bakuli and Smith [7], Lee et al. [8], Vassalos [9], Bendel and Klupfel [10], Chu et al. [11], and Cuesta et al. [12] provided snapshots of different ways in which variants of the evacuation routing problem can be analyzed and solved.

An emergency evacuation route is a sequence of movements of people away from the threat or actual occurrence of a hazard (such as fire, smoke, toxic clouds at ship etc) to a safe zone. Due to different group situations, some groups must be sent to some target safe points, but not the others (e.g., some groups must be sent to hospitals instead of police offices). In this paper we focus on the emergency evacuation routing problem with the aim to find the optimal assignment of the passenger groups to the evacuation tools. The evacuation processing time of a group by an evacuation tool is the routing time from the ship to the nearest safe point feasible to the group and returning the ship to rescue the next group. Specifically, we focus on a bi-criterion optimization problem. The primary objective is to minimize the evacuation time to save all the people during the evacuation operation, subject to a budget constraint. The secondary objective is to minimize the cost among all the alternative optimal solutions for the primary objective. In this paper we assume that all the given data are positive integers.

Along with search and rescue, the evacuation operation is the main component of disaster response by sea emergency managers. Urgent or emergency evacuation operations arise at short notice or without notice, and the speed of intervention in emergency planning and routing is critical to saving human lives and the environment. Evacuation operations are to be scheduled as rapidly as possible. This is a crucial issue addressed in this study.

II. A BRIEF REVIEW OF RELATED WORKS

It is worth noticing that emergency evacuation from large passenger ships has much in common with the problem of emergency evacuation from high-rise buildings, which has been extensively studied in the literature in recent years (see, e.g., [14-18], among many others). At the same time, in the mathematical models of evacuation at sea, we need to take into account the problem's conditions that are specific for passenger evacuation at sea, i.e., a very short time for the possible evacuation operation, usually not exceeding 30 minutes, before the ship sinks; and the dynamic characteristics of the problem, including such factors as bad weather and a stormy sea, while the ship at risk, as well as the lifeboats and saving vessels can be moving in the open sea.

Today there are about a dozen approaches to modelling emergency evacuation from ships at sea, differing in model scale, methodology, evacuation strategies, levels of detail etc. Depending on the model scale, the evacuation model may be microscopic, macroscopic or mesoscopic. In the *microscopic* model, the behaviour of each single passenger is modelled individually. In contrast, the *macroscopic* model is based on the behavioural analysis of large passenger groups and similarity of people group flow to flow of liquid or gas. The macroscopic model is advocated by the IMO Interim Guidelines [2]. Since neither of these two evacuation models alone can be fully appropriate for both high-level and low-level evacuation planning, the intermediate *mesoscopic* model is also used. This model serves to cover the gap between the microscopic and the macroscopic models [5]. We study the mesoscopic model in this paper.

From the standpoint of methodology, there are extensive studies on mathematical modelling of the evacuation-from-ship processes in emergency situations. In this work we pay attention to the quantitative and Operations Research-based scheduling approach.

As far as we know, the first mathematical model that treats the evacuation scheduling process and analyzes evacuee flows from the quantitative perspective was suggested by Togawa [19]. A mathematical evacuation model is a system of equations and/or inequalities that describe, simulate, and evaluate the process of evacuation from the ship, with the most significant factors describing the behaviours of the environment, ship, evacuees, and crew in egress situations being taken into account. Network flows, dynamic programming, and mixed integer programming are the main computational methods used to optimize the evacuation route. For example, researchers have formulated the evacuation problem as a minimum-cost network flow problem on the time-expanded network. Similar exact algorithms generate the optimal solutions, so they are attractive from the theoretical and practical points of view. In recent years, several computer programs for supporting emergency evacuation routing have been developed. Surveys of such studies can be found in Miah [20] and Hoffman et al. [21].

In such a model, the evacuation factors are usually defined under four categories, namely ship configuration, procedures, environment, and behaviour as follows (Lee et al. [8]): (a) ship configuration includes the effects on people behaviour of the ship structure, including exit widths, exit arrangement etc; (b) procedures include the knowledge of the passengers and crew, the training and activities of staff, and the familiarity of individual passengers with exit availability; (c) environment inside the structure describes the effects of heat, toxins, and smoke on the occupant's ability to navigate correctly and make right decisions; and (d) behaviour of the occupants is a crucial factor mapping the response of the individual and a group to the emergency, expected travel speeds, and the ability of the groups to carry out the desired actions.

The first computer program for evacuation analysis known to us is EVACNET, which was developed by Francis and Kisko [22] in the 1980s in FORTRAN 77. It was designed to compute and simulate evacuation times and define the distribution of moving evacuees. Specific marine oriented factors, such as ship listing, i.e., leaning to one side, and motion, crowd density, as well as psychological and behavioural factors affecting passenger behaviour in ship evacuation are taken into account and evaluated quantitatively.

In recent years, the emergency routing process has been extensively studied and successfully solved with the help of the network-flow models and methods. We refer to the following typical works in this research stream: Hamacher and Tjandra [15], Hoppe and Tardos [16], Yuan and Wang [24], and Caunhye et al. [25]. These works can be grouped into three main streams as follows:

- Exact deterministic models and methods that generate optimal evacuation plans maximizing the total number of saved people or minimizing the total evacuation time. The main algorithms used in this approach are based on dynamic programming, network flow analysis, and mathematical programming.
- Approximation and heuristic deterministic methods, such as the capacity-constrained routing heuristics (e.g., Elalouf et al. [26]).
- Simulation models of traffic flows with the help of computer experiments (Desmet and Gelenbe [27]).

To the best of our knowledge, the model we propose in this paper is new, and the suggested algorithm is original and the first of its kind in the literature.

III. PRELIMINARY RESULTS OF THE AUTHORS

In the recent works, Tang et al. [28] and Ng et al. [29] considered a real-time emergency evacuation problem that seeks to compute a set of rapid evacuation routes in a building. Given a three-dimensional geometric structure of the evacuation network, each room/exit is designated as a node and the corridors/links between the rooms as edges. The evacuation times are assigned to the edges. The authors provided a pseudo-polynomial-time dynamic programming algorithm to solve this problem, and constructed two types of approximation algorithm, namely a fully polynomial-time approximation scheme providing "almost-optimal" solutions and a fully polynomial-time approximately-feasible scheme, to yield a best "almost feasible" solution. A case study and the results of computational experiments to illustrate the working and efficacy of the proposed solution methods, respectively, were presented.

In [30] the authors considered the problem of searching and rescuing lost people during incidents in buildings and at sea. They presented a mathematical model for the search-and-rescue process and a new fast scheduling algorithm for solving the problem.

In [13], the authors considered the problem of non-preemptive scheduling of a set of independent jobs (e.g., groups of people) on a set of uniform parallel machines (e.g., people-saving tools), where each job has a set of machines to which it can be assigned. This kind of restriction is called the *processing set restriction*. In the literature there are many types and applications of the processing set restrictions that have been studied. Leung and Ng [13] considered two kinds: the "inclusive processing set" and the "tree-hierarchical processing set". They provided fast approximation algorithms for both cases and showed that they both have a worst-case performance bound of 4/3. Moreover, they showed that the bounds are achievable. For the completeness of discussion, let us consider the Leung-Ng model and algorithm in more detail.

For ease of reference, we summarize the notation used in this session as follows:

n = number of independent jobs;
$J = \{J_1, J_2, ..., J_n\}$ is the set of n jobs;
m = number of uniform parallel machines;
$M = \{M_1, M_2, ..., M_m\}$ is the set of m uniform parallel machines;
s_i = speed of machine M_i (without loss of generality, assume $1 = s_1 \leq s_2 \leq \ldots \leq s_m$);
z = number of distinct speeds in s_i, $i = 1, \ldots, m$;
p_j = processing time of job J_j when it is processed by a machine of speed 1;
$P = \sum p_j$ is the sum of all the processing times p_j;
MS_j = the set of machines to which job J_j can be assigned;

$C_{\max}$ = makespan, i.e., the total processing time;
LB = a lower bound on $C_{\max}$;
UB = an upper bound on $C_{\max}$;
C = an input parameter for testing whether the jobs in J can be scheduled on the machines in M so that the makespan is not larger than $4C/3$;
JB_k, $k = 1, \ldots, z$, form a partition of J; and MA_k, $k = 1, \ldots, z$, form a partition of M, such that, each of the $k = 1, \ldots, z$, jobs in JB_k can only be processed by any one machine in $MA_k \cup \cdots \cup MA_z$;
$MA_k = \{M_{a[k]}, M_{a[k]+1}, \ldots, M_{b[k]}\}$, i.e., $M_{a[k]}$ and $M_{b[k]}$ are the first and last machines in MA_k, $k = 1, \ldots, z$;
$W = \{k/(4\text{LCM}(s_1,\ldots,s_z)) : k \text{ is a positive integer}\}$ is the set of all the positive multiples of $1/(4\text{LCM}(s_1,\ldots,s_z))$, where $\text{LCM}(s_1,\ldots,s_z)$ is the least common multiple of $s_1, \ldots, s_z$.

Leung and Ng [13] considered the problem of non-preemptively scheduling n independent jobs $J = \{J_1, J_2, \ldots, J_n\}$ on m uniform parallel machines $M = \{M_1, M_2, \ldots, M_m\}$, where the machines differ in speed but not in functionality. Each job J_j has a processing time p_j and each machine M_i has a speed s_i. Without loss of generality, we assume that $1 = s_1 \leq s_2 \leq \ldots \leq s_m$. If job J_j is processed by machine M_i, then it takes p_j/s_i units of real time to complete. Each job J_j has a set of machines $MS_j \subseteq M$ to which it can be assigned. The goal is to find a schedule so that each job J_j is assigned to one of the machines in MS_j and that the makespan, i.e., the total processing time $C_{\max}$, is minimized. In terms of evacuation operations, the total time of evacuating all the saved people should be minimized.

The authors studied the classes of "inclusive processing set" and "tree-hierarchical processing set" restrictions. The "inclusive processing set" restriction has the property that for each pair of jobs J_i and J_j, either $MS_i \subseteq MS_j$ or $MS_j \subseteq MS_i$. The "tree-hierarchical processing set" has the property that each machine M_k is associated with a vertex of a tree, and that the processing set of a job J_j is the set of machines composed of its associated vertex and all the vertices on the unique path from its associated vertex to the root of the tree.

The Leung-Ng algorithm computes a lower bound LB and an upper bound UB on the optimal makespan, and conducts a binary search in the interval [LB, UB]. For each value C encountered in the binary search, the algorithm tries to schedule the jobs in J on the machines in M so that the makespan is not larger than $4C/3$. If it is successful, then the search is conducted in the lower half of the range; otherwise, the search is conducted in the upper half of the range.

We briefly describe the Leung-Ng procedure [13] to test if a set of jobs can be scheduled on the machines so that the makespan is not larger than $4C/3$ for any given input parameter C.

The subroutine, denoted by Test(C), that schedules the jobs takes as input an integer parameter C. It returns "Yes" if it is possible to schedule all the jobs so that the makespan is $4C/3$ or less; otherwise, it returns "No". For a given value of C, Leung and Ng [13] called job J_j a Type-1 job with respect to machine M_i of speed s_i if its processing requirement p_j/s_i is more than $2C/3$. They called J_j a Type-2 job if its processing requirement p_j/s_i is more than $C/3$ but not more than $2C/3$. Finally, they called J_j a Type-3 job if its processing requirement p_j/s_i is not more than $C/3$. Due to the "inclusive processing set" restriction, the jobs and machines can be classified into z groups, i.e., JB_k and MA_k, $k = 1, \ldots, z$ such that, for each $k = 1, \ldots, z$, jobs in JB_k can only be processed by any one machine in $MA_k \cup \cdots \cup MA_z$. The subroutine schedules the jobs in JB_k on the machines in MA_k from $k = 1$ until $k = z$. For each k, the subroutine assigns jobs in JB_k to the machines in MA_k in the order from $M_{a[k]}$ to $M_{b[k]}$ as follows:

Suppose we are considering machine M_j, $a[k] \leq j \leq b[k]$. If JB_k has one or more Type-1 jobs, then schedule the longest Type-1 job on M_j. Otherwise, if JB_k has two or more Type-2 jobs, then schedule the two longest Type-2 jobs on M_j. Otherwise, if JB_k has only one Type-2 job, then schedule the Type-2 job on M_j. After this, schedule the Type-3 jobs, in any order, until either there is no more Type-3 job or encountering a situation where the scheduling of the job will make the makespan greater than $4C/3$, whichever occurs first. The scheduled jobs will be deleted from JB_k. If JB_k becomes the empty set, then we repeat the above process to schedule the jobs in JB_{k+1} on the machines in MA_{k+1}. Otherwise, we consider the machine M_{j+1}. If $j+1 = a[k+1]$, then we merge JB_k with JB_{k+1} before scheduling any job. Leung and Ng [13] proved that the running time of their algorithm is $O(mn \log P)$, where $P = \sum p_j$.

IV. A MODIFICATION OF THE LEUNG-NG ALGORITHM

The Leung-Ng algorithm [13] provides a sub-optimal solution to the $Q \mid M_j(inclusive) \mid C_{max}$ problem in $O(mn \log P)$ time with a worst-case bound of 4/3. In this section we provide a modification of this algorithm to obtain an optimal algorithm for the $Q \mid M_j(inclusive) \mid C_{max}$ problem.

Let $W = \{k/(4\mathrm{LCM}(s_1,...,s_z)) : k \text{ is a positive integer}\}$ be the set of all the positive multiples of $1/(4\mathrm{LCM}(s_1,...,s_z))$, where $\mathrm{LCM}(s_1,...,s_z)$ is the least common multiple of $s_1, \ldots, s_z$. We have the following lemma.

Lemma 1

If there is only one element $w \in W$ in an interval $[a,b]$, and Test(c) returns "No" for all $0<c<a$ while Test(b) returns "Yes", then the schedule given by Test(b) is optimal to $Q \mid M_j(inclusive) \mid C_{max}$ and the optimal makespan equals $4w/3$.

Proof.

Let $W_1 = \{k/\mathrm{LCM}(s_1,...,s_z) : k \text{ is a positive integer}\}$ be the set of all the positive multiples of $1/\mathrm{LCM}(s_1,...,s_z)$. Any processing time must be in the form of $p_j/s_i \in W_1$. Therefore, the completion time of any schedule must be in W_1. Under the assumptions of Lemma1, if there is $a \leq x<w$ such that Test(x) returns "Yes" with makespan $u \in W_1$, then let $x_1=3u/4 \in W$. Since $u \leq 4x/3$, we have $x_1 \leq x<w$. Because w is the only element in $W \cap [a,b]$, we immediately get $x_1<a$. This means there is a schedule with makespan $u= 4x_1/3$ and $x_1<a$, which is a contradiction.

Let the makespan of the schedule returned by Test(b) be $u_1 \in W_1$. We further let $x_2=3u_1/4 \in W$. Since $u_1 \leq 4b/3$, we have $x_2 \leq b$. Test(x_2) must return "Yes", as the schedule returned by Test(b) is one with makespan = $4x_2/3$. Hence, $w \leq x_2 \leq b$, so $x_2=w$. As Test(w) returns "Yes" and Test(c) returns "No" for all $0<c<w$, the result follows. □

Following Leung and Ng [13], we compute LB = $\min\{L_k : 0 \leq k \leq z\}$, where $L_0 = p_{\max}/s_z$, $p_{\max} = \max\{p_j : 1 \leq j \leq n\}$, and $L_k = \left(\sum_{J_j \in JB_k \cup \cdots \cup JB_z} p_j\right) / \left(\sum_{j=k}^{z} m_j s_j\right)$, $1 \leq k \leq z$. For UB, we take UB = P/s_z.

Since the optimal makespan must be within [LB, UB], it suffices to consider all C∈[3LB/4, 3UB/4] as input to Test(C) to find the optimal makespan. Note that Test(C) returns "No" for all 0<C<3LB/4 while Test(3UB/4) returns "Yes". We can make use of the bisection method for narrowing the interval to be considered. That is, in each iteration, we consider an interval [a, b] with Test(C) returning "No" for all 0<C<a while Test(b) returning "Yes". Then, we let v = (a+b)/2. If Test(v) returns "No", then the next iteration to consider is [v, b]. If Test(v) returns "Yes", then the next iteration to

consider is [a, v]. Continue in this way until we have an interval with width < $1/\mathrm{LCM}(s_1,...,s_z)$. Then, by Lemma 1, we obtain the optimal solution. The algorithm is as follows:

Modified Leung-Ng Algorithm

1. Set a := LB, d := UB.
2. While d-$a \geq 1/\mathrm{LCM}(s_1,...,s_z)$, do
   (a) Set v:= (a+b)/2.
   (b) If Test(v) returns "No", then set a := v.
   else if Test(v) returns "Yes", then set d:= v.
3. The schedule π and makespan MC given by Test(d) are optimal (by Lemma 1).
4. Output π and MC.

Lemma 2

The modified Leung-Ng algorithm gives an optimal solution to Q | M_j(inclusive) | C_{max} in O((log P+ n log $s_{\max}$)nm) time, where $s_{\max}$= max$\{s_i : 1 \leq i \leq z\}$.

Proof.

It is evident that the modified Leung-Ng algorithm gives an optimal solution to Q | M_j(inclusive) | C_{max}. It calls Test(m) for O(log((UB – LB) $\mathrm{LCM}(s_1,...,s_z)$)) ≤ O(log(($P/s_z$)($s_1 \ldots s_z$))) = O(log($Ps_1 \ldots s_{z-1}$))) ≤ O(log P + n log $s_{\max}$) time. Since Test(v) takes O(nm) time, the overall algorithm runs in O((log P + n log $s_{\max}$)nm) time. □

V. A New Evacuation Scheduling Algorithm

In this section we propose a new evacuation scheduling algorithm, which iteratively calls the modified Leung-Ng algorithm in Section IV for finding the optimal evacuation schedule for all the people from a ship in an accident at sea. Suppose that n groups of people are trapped on different floors of the ship during an accident, e.g., fire, waiting for evacuation teams to take them from the ship to a safe place. The groups are known to be different in their at-risk levels, according to their age, gender, and closeness to the place where the accident happens.

We consider a rather common situation where the evacuation teams can exploit two types of evacuation tools, namely helicopters and life-saving boats, to save the groups of people. While the boats can be used to save the low-risk people and groups, some high-risk groups can only be saved by helicopters. This implies an "inclusive processing set" restriction discussed in the previous section. Comparatively, helicopters are fast but expensive, while boats are slow but cheap.

Suppose that m_1 boats and m_2 helicopters are to be used for the evacuation. Let c_1 and c_2 be the unit costs of a boat and a helicopter, respectively. Clearly, $c_1 < c_2$. Naturally, we have $c_1m_1 + c_2m_2 \leq b$, where b (> c_1) is the available budget. Let p_j (j = 1, …, n) be the processing time to take group j from the ship to a safe place by using saving tool 1; s_i be the speed of saving tool i (i = 1, …, m_1 for boats, i = m_1+1, …, m_1+m_2 for helicopters). Following Leung and Ng [9], we assume that s_1=1. Hence, $1 = s_1 = \ldots = s_{m_1} < s_{m_1+1} = \ldots = s_{m_1+m_2}$. Therefore, the processing time to save group j by saving tool i is p_j/s_i. Suppose that groups 1, 2, …, n_1 can be saved by either a boat or a helicopter, and groups n_1+1, n_1+2, …, n can only be saved by a helicopter. This gives rise to a processing set restriction. An evacuation schedule π defines which group of people is assigned to which saving tool, and in which order. Our objective is to find the quantities of saving tools m_1 and m_2, and the evacuation schedule for all the groups to minimize the evacuation makespan as the first priority and the budget as the second priority, subject to a budget constraint.

Mathematically, the bi-criterion optimization problem is as follows:

Goal 1: Minimizing the makespan by finding an optimal schedule σ, and m_1 and m_2, subject to a budget constraint and processing set constraints.

$$\underset{\sigma, m_1, m_2}{Min} \; C_{\max}(\sigma)$$

subject to

$$c_1 m_1 + c_2 m_2 \leq b$$
$$MS_j = MA_1, j \in \{1, \ldots, n_1\}$$
$$MS_j = MA_2, j \in \{n_1+1, n_1+2, \ldots, n\},$$

where $MA_1 = \{M_1, M_2, \ldots, M_{m_1+m_2}\}$ and $MA_2 = \{M_{m_1+1}, M_{m_1+2}, \ldots, M_{m_1+m_2}\}$.

Goal 2: Minimizing *the* total cost (among all the optimal solutions in Goal 1, with minimum makespan $C^*_{\max}$)

$$\underset{m_1, m_2}{Min} \; c_1 m_1 + c_2 m_2$$

subject to

$$C_{\max}(\sigma) = C^*_{\max}$$

and

constraints of Goal 1.

We denote the modified Leung-Ng algorithm with m_1 boats and m_2 helicopters in the above setting by $MLN(m_1, m_2)$. Let $\lfloor x \rfloor$ denote the largest integer less than or equal to x. Then we define a new iterative algorithm as follows:

<u>Evacuation Scheduling Algorithm</u>

Finding the minimum C_{max}

1. Set $T := P$.
2. For $m_2 = 0$ to $\lfloor b/c_2 \rfloor$, do
    (a) Set $m_1 := \lfloor (b - c_2 m_2)/c_1 \rfloor$.
    (b) Run $MLN(m_1, m_2)$ to find schedule π and the optimal makespan $C_{\max}$ for (m_1, m_2).
    (c) Set $CM(m_2) := C_{\max}$.
    (d) If $T \geq C_{\max}$, then set $T := C_{\max}$ and $\sigma := \pi$.
3. Output T and σ.

Finding the minimum cost

4. Set $S := b$, $\theta := b$, $m_{1*} := 0$, $m_{2*} := 0$, $\pi := [1,2,\ldots,n]$.
5. For $m_2 = 0$ to $\lfloor b/c_2 \rfloor$, do

(a) If CM(m_2) = T, then
   (a1) Set $m_1 := \lfloor (b - c_2 m_2)/c_1 \rfloor$.
   (a2) Run BS(m_1,m_2) to find the smallest m_1 with the same optimal makespan, schedule π, and budget θ (see below).
   (a3) If $S \geq \theta$, then set $m_{1*} := m_1$, $m_{2*} := m_2$, $S := \theta$, and $\sigma_1 := \pi$.
6. Output m_{1*}, m_{2*}, S, T, and σ_1.

The idea for the algorithm is as follows: We search for all the possible (integer) values of m_2 in the interval [0, b/c_2]. For each m_2, we take the maximum value of m_1 within the budget constraint to obtain the minimum makespan for the given m_2 (see Figure 1). Then, we take the minimum of all such values to obtain the overall minimum makespan of the problem.

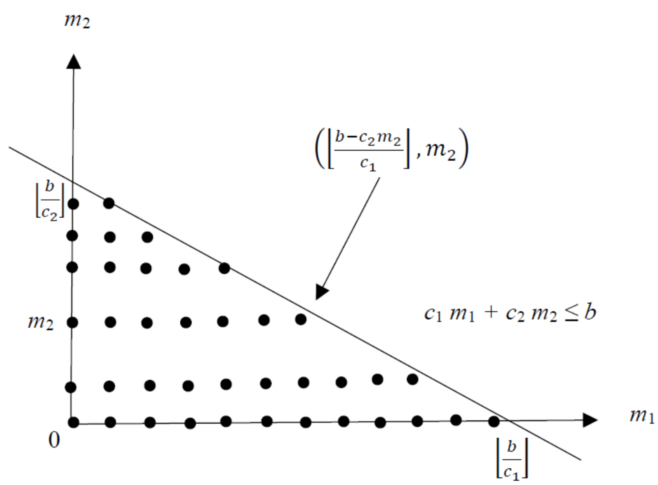

Figure 1 Integer values of (m_1, m_2) under the budget constraint

To find the minimum cost under the minimum makespan, we do a binary search for each m_2 in {0, …, $\lfloor b/c_2 \rfloor$} such that CM(m_2) = T, i.e., the minimum makespan. Then, we find the minimum of all these costs.

Binary search algorithm BS(m_1,m_2)
1. Set a := 0, d := m_1.
2. While d-a ≥ 1, do
   (a) Run *MLN*(m_1, m_2) to find schedule π' and the optimal makespan $C_{\max}$ for (m_1, m_2).
   (b) If $C_{\max} > T$ and d-a > 1, then set a := m_1
      else if $C_{\max} > T$ and d-a = 1, then set a := d
         else set d := m_1 and $\pi = \pi'$.
   (c) Set $m_1 := \left\lfloor \frac{a+d}{2} \right\rfloor$.
3. Set $\theta := c_1\, m_1 + c_2\, m_2$.
4. Output m_1, m_2, π, and θ.

We obtain the following result.

Theorem 1
The Evacuation Scheduling Algorithm can find an optimal solution to the evacuation problem.

Consider now the time complexity of the Evacuation Scheduling Algorithm. For *finding the minimum* C_{max}, we can show that it is polynomial in b. Indeed, since we run the modified Leung-Ng algorithm for $\lfloor b / c_2 \rfloor$ times and the running time of $MLN(m_1, m_2)$ is $O((\log P + n \log s_{\max}) * (m_1 + m_2)n) \le O((\log P + n \log s_{\max}) * bn / c_1)$, the overall running time is $O((\log P + n \log s_{\max}) * b^2 n / c_1 c_2)$. For *finding the minimum cost*, since we run the bisection algorithm for at most $\lfloor b / c_2 \rfloor$ times and the running time of $BS(m_1, m_2)$ is $O(\log(b/c_1) * (\log P + n \log s_{\max}) * bn / c_1)$, the overall algorithm runs in $O(\log(b/c_1) * (\log P + n \log s_{\max}) * b^2 n / c_1 c_2)$ time. Hence, we have the following theorem.

Theorem 2
The running time of the Evacuation Scheduling Algorithm is at most $O(\log(b/c_1) * (\log P + n \log s_{\max}) * b^2 n / c_1 c_2)$.

Remark: The above algorithm can be easily extended to $k \ge 1$ types of evacuation tools, having m_r evacuation tools for each type ($r = 1, \ldots, k$) under the budget constraint $c_1 m_1 + \cdots + c_k m_k \le b$. The running time of the resulting algorithm is at most $O(\log(b/c_1) * (\log P + n \log s_{\max}) * b^k n / c_1 c_2 \cdots c_k)$.

VI. Example

Consider the following example, in which there are five groups of evacuees to be saved by boats and/or helicopters (see Tables 1 and 2).

Table 1. Costs and budget.

	Unit cost (in US$1,000)	Speed (s_i)
Boat (B)	2	1
Helicopter (H)	5	2
Budget	15	

Table 2. Operation durations and saving facilities for each job.

Group	p_j (in minutes)	Set of feasible evaluation types
1	10	{B, H}
2	20	{B, H}
3	30	{B, H}
4	10	{H}
5	20	{H}

The Evacuation Scheduling Algorithm first finds the minimum makespan as presented in Table 3.

Table 3. The minimum makespan found by the algorithm.

m_2	$m_1 = \left\lfloor \frac{15-5m_2}{2} \right\rfloor$	C_{max} (in minutes)
0	7	---*
1	5	30
2	2	20
3	0	20

*: No feasible schedule since groups 4 and 5 must be saved by H.

This table shows that the minimum makespan is 20 minutes at (m_1, m_2) = (2,2) or (0,3). To obtain the minimum cost, the Evacuation Scheduling Algorithm makes use of binary search iteratively to obtain the following results (see Table 4).

Table 4. The minimum costs.

m_2	m_1^*	C_{max} (in minutes)	Cost (in US K$)
2	1	20	12
3	0	20	15

Therefore, $m_1^* = 1$, $m_2^* = 2$, optimal cost = US$12,000, and optimal makespan = 20 minutes with the following optimal schedule (see Table 5).

Table 5. The optimal assignments of the facilities.

Assignment	Completion time (in minutes)
J_1--> M_1	10
J_2,J_5--> M_3	20
J_3,J_4--> M_2	20

VII. COMPUTATIONAL EXPERIMENTS

We coded the proposed algorithm in MATLAB 7.8.0. (R2009a) and conducted all the computational tests using a Microsoft Windows 10 Home Basic system, which comprises an x64-based PC equipped with an Intel ® Core ™ i5-3317U processor with speed at CPU@1.70GHZ and memory of RAM 4.00GB. We computationally tested the performance of the proposed algorithm for solving to optimality 1,000 numerical examples generated randomly. In the examples, we took five groups of evacuees to be saved by two types of saving equipment, say, boats or helicopters. We considered a multi-parametric data vector $\boldsymbol{D} = (\boldsymbol{p}, c_1, c_2)$, where $\boldsymbol{p} = (p_1, p_2, p_3, p_4, p_5)$, and each of the three components of the vector $\boldsymbol{D}$ randomly takes ten possible values. Therefore, we executed our algorithm on MATLAB-4 for a total of 1,000 possible samples of the available values for the random vector $\boldsymbol{D}$. Each of the p_j values was generated uniformly from the interval [6, 40]. Different configurations of $\boldsymbol{p}$ were generated ten times and run together with different combinations of (c_1, c_2), where the c_1 and c_2 values were independently generated ten times each from the intervals [2, 5] and [10, 20], respectively. Reporting the computational results, Table 6 presents one of the ten possible combinations of the p_j values and Table 7 presents the sample cost values. In addition, Table 8 reports the optimal costs and optimal assignments of facilities for ten randomly selected sample problems.

Table 6. A sample configuration of the p_j values.

	p_j (in minutes), for ten runs										Set of feasible evaluation types
Jobs J_i	1	2	3	4	5	6	7	8	9	10	
1	10	16	30	24	28	28	18	24	12	24	$\{B,H\}$
2	8	28	8	12	28	24	16	30	10	20	$\{B,H\}$
3	16	28	28	14	20	18	22	24	40	16	$\{B,H\}$
4	24	20	12	12	20	32	16	22	14	24	$\{H\}$
5	16	20	24	28	32	20	12	16	6	30	$\{H\}$

Table 7. The costs and budget for 10 randomly selected sample problems.

The sample number	1	2	3	4	5	6	7	8	9	10
Boat cost (in US$1,000)	3	4	2	5	3	2	3	2	4	3
Helicopter cost (in US$1,000)	21	20	13	19	27	11	21	19	20	15
Speed of a boat	1	1	1	1	1	1	1	1	1	1
Speed of a helicopter	2	2	2	2	2	2	2	2	2	2
Budget (US$1,000)	84	79	74	100	120	50	70	76	70	80

Table 8. The optimal costs and optimal assignments of facilities for ten sample problems (m_1 and m_2 stand for the number of helicopters and the number of boats, respectively).

Case number	m_2	m_1	C_{max} (in minutes)	Unit cost (in US$1,000)	*Assignment*
1	2	1	17	45	$J_1,J_4 \to H, J_2 \to B, J_3,J_5 \to H$
2	2	1	24	46	$J_1 \to B, J_2,J_4 \to H, J_3,J_5 \to H$
3	3	1	17	41	$J_2 \to B, J_1,J_4 \to H, J_3,J_5 \to H$
4	3	2	14	61	$J_1 \to H, J_2 \to B, J_3 \to B, J_4 \to H, J_5 \to H$
5	4	0	20	108	$J_1 \to H, J_2 \to H, J_3,J_4 \to H, J_5 \to H$
6	4	0	21	44	$J_1 \to H, J_2,J_3 \to H, J_4 \to H, J_5 \to H$
7	2	1	16	45	$J_1,J_3 \to H, J_2 \to B, J_4,J_5 \to H$
8	4	0	19	76	$J_1 \to H, J_2 \to H, J_3 \to H, J_4,J_5 \to H$
9	2	2	20	48	$J_1 \to B, J_2 \to B, J_3 \to H, J_{4,}J_5 \to H$
10	5	0	15	75	$J_1 \to H, J_2 \to H, J_3 \to H, J_4 \to H, J_5 \to H$

The total running time for solving all 1,000 examples to optimality is 14.5 minutes, i.e., less than 1 second for solving one sample problem, on average. The computational results demonstrate the practical applicability of the proposed algorithm.

VIII. CONCLUSION

In this paper we design a fast algorithm to solve the scheduling problem arising in emergency evacuation of different groups of people from ships. With suitably defined evacuation tools, the devised algorithm can be applied to other similar emergency evacuation situations, e.g., high-rise buildings and hills. Although the algorithm is very fast in practice, which runs in polynomial time (in b), it is classified as *pseudo-polynomial* because the running time is a polynomial function of b, not of $\log b$.

An interesting direction for future research is to devise a (strongly or weakly) polynomial algorithm for the considered scheduling problem. We believe that the binary search technique adopted by Leung and Ng in [13] could be a viable way for achieving this aim. Another interesting direction is to consider the floating/moving nature of the ship, which makes the evacuation processing times stochastic or fuzzy. Besides, extending the analysis to multi-criterion rescue and evacuation is also an important future research direction.

ACKNOWLEDGMENTS

We thank two anonymous referees for their helpful comments on earlier versions of our paper. This research was supported in part by the Research Grants Council of Hong Kong under grant numbers PolyU 152148/15E and PolyU 152207/17E.

References

[1] FCCA, Florida-Caribbean Cruise Association, *Cruise Industry Overview 2018: State of the Cruise Industry*. Pembroke Pines, Florida, 2018, 12 pp.

[2] IMO, International Marine Organization, *Interim Guidelines for Evacuation Analysis of New and Existing Passengers Ships*, MSC/Circ.1033, June 2002.

[3] D. Mishra, S. Kumar and E. Hassini, “Current trends in disaster management simulation modelling research”. *Annals of Operations Research*, 283(1-2),1387-1411, 2019.

[4] C. Duhamel, A.C. Santos, D. Brasil, E. Châtelet and B. Birregah, “Connecting a population dynamic model with a multi-period location-allocation problem for post-disaster relief operations”. *Annals of Operations Research*, 247(2), 693-713, 2016.

[5] U. Pyakurel and T.N. Dhamala, “Continuous dynamic contraflow approach for evacuation planning”. *Annals of Operations Research*, 253(1), 573-598, 2017.

[6] A. Akbari, R. Pelot and H.A. Eiselt, “A modular capacitated multi-objective model for locating maritime search and rescue vessels”. *Annals of Operations Research*, 267(1-2), 3-28, 2018.

[7] D.L. Bakuli and J.M. Smith, “Resource allocation state emergency evacuation networks”. *European Journal of Operational Research*, 89, 543-553, 1996.

[8] D. Lee, H. Kim, J.-H. Park and B.J. Park, “The current status and future issues in human evacuation from ships”. *Safety Science*, 41, 861-876, 2003.

[9] D. Vassalos, “Passenger ship safety: Containing the risk”. *Marine Technology*, 43(4), 203-212, 2006.

[10] J. Bendel and H. Klüpfel, “Accessibility and evacuation planning – similarities and differences”. *Pedestrian and Evacuation Dynamics*, 701-712, 2011.

[11] C.W. Chu, H.A. Lu and C.Z. Pan, “Emergency evacuation route for the passenger ship”. *Journal of Marine Science and Technology*, 21(5), 515-521, 2013.

[12] A. Cuesta, D. Alvear, D. Abreu and D.Silio, “Real-time stochastic evacuation models for decision support in actual emergencies”. *Proceedings of the 11th International Symposium on Fire Safety Science* (SFSS-14), February 2014, University of Canterbury Press, New Zealand, 10-14, 2014.

[13] J.Y-T. Leung and C.T. Ng, “Fast approximation algorithms for uniform machine scheduling with processing set restrictions”. *European Journal of Opererational Research*, 260, 507-513, 2017.

[14] W. Choi, H.W. Hamacher and S. Tufecki, “Modeling of building evacuation problems by network flows with constraints”. *European Journal of Operational Research*, 35, 98-110, 1988.

[15] H.W. Hamacher and S.A. Tjandra, “Mathematical modeling of evacuation problems: A state of the art”. *Pedestrian and Evacuation Dynamics*, 227-266, 2002.

[16] B. Hoppe and E. Tardos, “Polynomial time algorithms for some evacuation problems”. *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, 433-441, 1994.

[17] P.B. Luh, C.T. Wilkie, S.-C. Chang, K.L. Marsh and N. Olderman, “Modeling and optimization of building emergency evacuation considering blocking effects on crowd movement”. *IEEE Transactions on Automation Science and Engineering*, 9(4), 687-700, 2012.

[18] M.S. Osman and B. Ram, “Two-phase evacuation route planning approach using combined path networks for building and roads”. *Computers & Industrial Engineering*, 65(2), 233-245, 2013.

[19] K. Togawa, *Study of Fire Escapes Basing on the Observation of Multitude Currents, Report No. 14,* Building Research Institute, Ministry of Construction, Tokyo, 1955.

[20] M. Miah, “Survey of data mining methods in emergency evacuation planning”. *Proceedings of the Conference for Information Systems Applied Research (CONISAR 2011)*, Wilmington North Carolina, USA, 2011.

[21] P. Hoffmann, M. Höferlin, A. Kirstädter and D. Weiskopf, “Dynamic evacuation planning by visual analytics — an expert survey”. *SIGRAD* 2013.

[22] R.L. Francis and T.M. Kisko, “EVACNET+: A computer program to determine optimal building evacuation plans”. *Fire Safety Journal*, 9, 211-220, 1985.

[23] D. Vassalos, G. Christiansen, H.S. Kim, M. Bole and J. Majumder, *Evacuability of Passenger Ships at Sea*, Evacuation Group of the Ship Stability Research Centre, 2001.

[24] Y. Yuan and D. Wang, “Path selection model and algorithm for emergency logistics management”. *Computers & Industrial Engineering*, 56(3), 1081-1094, 2009.

[25] A.M. Caunhye, X. Nie and A. Pokharel, “Optimization models in emergency logistics: A literature review”. *Socio-Economic Planning Science*, 46(1), 4-13, 2012.

[26] A. Elalouf, E. Levner and T.C.E. Cheng, “Routing and dispatching of mobile agents in integrated enterprises”. *International Journal of Production Economics*, 145(1), 96-106, 2013.

[27] A. Desmet and E. Gelenbe, “Graph and analytical models for emergency evacuation”. *Future Internet*, 5, 46-55, 2013.

[28] H. Tang, A. Elalouf, E. Levner and T.C.E. Cheng, “Efficient computation of evacuation routes on a three-dimensional geometric network”. *Computers & Industrial Engineering*, 76, 231-242, 2014.

[29] C.T. Ng, T.C.E. Cheng, A. Elalouf and E. Levner, “Design and analysis of a fast approximation algorithm for multi-modal emergency evacuation routes in the 3D environment”. In D. Chan (ed.). *Environmental Science and Information Application Technology*, CRC Press, 307-312, 2015.

[30] B. Kriheli, E. Levner, T.C.E. Cheng and C.T. Ng, “Scheduling of rescue operations with overlooking within short response time: An approach for quick response to urgent relief demand”. *International Journal of Advances in Electronics and Computer Science*, 4(11), 36-39, 2017.