The Mixed Evacuation Problem

Yosuke Hanawa[1], Yuya Higashikawa[2,3], Naoyuki Kamiyama[4,5(✉)],
Naoki Katoh[3,6], and Atsushi Takizawa[3,7]

[1] Kyoto University, Kyoto, Japan
yosuke.hanawa@gmail.com
[2] Chuo University, Tokyo, Japan
higashikawa.874@g.chuo-u.ac.jp
[3] JST, CREST, Saitama, Japan
[4] Kyushu University, Fukuoka, Japan
kamiyama@imi.kyushu-u.ac.jp
[5] JST, PRESTO, Saitama, Japan
[6] Kwansei-Gakuin University, Hyogo, Japan
naoki.katoh@gmail.com
[7] Osaka City University, Osaka, Japan
takizawa@arch.eng.osaka-cu.ac.jp

Abstract. A dynamic network introduced by Ford and Fulkerson is a
directed graph with capacities and transit times on its arcs. The quick-
est transshipment problem is one of the most fundamental problems in
dynamic networks. In this problem, we are given sources and sinks. Then,
the goal of this problem is to find a minimum time limit such that we
can send exactly the right amount of flow from sources to sinks. In this
paper, we introduce a variant of this problem called the mixed evacuation
problem. This problem models an emergent situation in which people can
evacuate on foot or by car. The goal is to organize such a mixed evacu-
ation so that an efficient evacuation can be achieved. In this paper, we
study this problem from the theoretical and practical viewpoints. In the
first part, we prove the polynomial-time solvability of this problem in
the case where the number of sources and sinks is not large, and also
prove the polynomial-time solvability and computational hardness of its
variants with integer constraints. In the second part, we apply our model
to the case study of Minabe town in Wakayama prefecture, Japan.

1 Introduction

The coastal area facing the Pacific Ocean in Japan ranging from Shizuoka pre-
fecture to Miyazaki prefecture has a high risk of a tsunami. In particular, it

is predicted that Nankai Trough Earthquake will occur with 70 % probability within thirty years, and it will trigger a tsunami of the huge size which will quickly arrive at the coast (see, e.g., [1]). Based on several assumptions and estimated data, Wakayama prefecture recently designated several areas in which it is difficult for all people in the area to evacuate to safety places such as tsunami evacuation buildings before a tsunami arrives when Nankai Trough Earthquake occurs. For example, it is predicted that in Kushimoto town located at the south end of the main land of Japan, a tsunami arrives at earliest within ten minutes. One of assumptions the prefecture used is that the evacuation is done only by walking. In principle, it used to be not allowed to use cars for evacuation because the usage of cars in such an emergent situation may block evacuation of pedestrians which was observed at the time of Tohoku-Pacific Ocean Earthquake. However, if it is allowed to use cars and the smooth evacuation by car is organized, then the evacuation completion time may be shortened. The aim of this paper is to propose a mathematical model for making such a good "mixed" evacuation plan.

In this paper, we use a dynamic network flow introduced by Ford and Fulkerson [2,3] for modeling such a mixed evacuation. A dynamic network is a directed graph with capacities and transit times on its arcs. The quickest transshipment problem is one of the most fundamental problems in dynamic networks. In this problem, we are given a dynamic network with several sources and sinks. Furthermore, we are given a supply for each source and a demand for each sink. Then, the goal of this problem is to find the minimum time limit such that we can send exactly the right amount of flow from sources to sinks. Hoppe and Tardos [4] proved that this problem can be solved in polynomial time. In this paper, we introduce a variant of the quickest transshipment problem called the mixed evacuation problem. This problem models an emergent situation in which people can evacuate on foot or by car. The goal of this problem is to organize such a mixed evacuation so that an efficient evacuation can be achieved. In the first part of this paper, we study the mixed evacuation problem from the theoretical viewpoint. First we prove that if the number of sources and sinks is at most $C \log_2 n$ (n is the number of vertices) for some constant C, then mixed evacuation problem can be solved in polynomial time (Sect. 3). In addition, we consider variants of the mixed evacuation problem with integer constraints (Sect. 4). In the second part of this paper, we study the mixed evacuation problem from the practical viewpoint. In this part, we apply our model to the case study in Japan (Sect. 5). More precisely, we apply our model for Minabe town in Wakayama prefecture, which was designated as a city in which safe evacuation from a tsunami is difficult when Nankai Trough Earthquake occurs.

Let $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$, $\mathbb{Z}_+$, and $\mathbb{Z}_-$ be the sets of reals, non-negative reals, integers, non-negative integers, and non-positive integers, respectively. For each finite set U, each vector x in $\mathbb{R}^U$, and each subset W of U, we define $x(W) := \sum_{u \in W} x(u)$. Furthermore, for each finite set U and each pair of vectors x, y in $\mathbb{R}^U$, we define $\langle x, y \rangle := \sum_{u \in U} x(u)y(u)$.

2   Preliminaries

The MIXED EVACUATION problem is defined as follows. We are given a directed graph $D = (V, A)$ and two disjoint subsets S, T of V. Define $n := |V|$. The subset S (resp., T) is the set of source vertices (resp., sink vertices) in V. We assume that no arc in A enters (resp., leaves) a vertex in S (resp., T). In addition, we are given an arc capacity vector c in $\mathbb{Z}_+^A$, a supply vector b in $\mathbb{Z}_+^S$, a sink capacity vector u in $\mathbb{Z}_-^T$, transit time vectors τ_1, τ_2 in $\mathbb{Z}_+^A$, and fluid coefficients q_1, q_2 in $\mathbb{Z}_+$. In our application, τ_1 represents the speed of walking, and q_1 represents the number of people that can walk in the one unit of the arc capacity. The values τ_2, q_2 represent the information for cars. Lastly, we are given a time limit Θ in $\mathbb{Z}_+$. Define $[\Theta] := \{0, 1, \ldots, \Theta\}$.

For each integer i in $\{1, 2\}$, each function $f : A \times [\Theta] \to \mathbb{R}_+$, each vertex v in V, and each integer θ in $[\Theta]$, we define

$$\partial_i f(v, \theta) := \sum_{a \in \delta(v;A)} \sum_{t=0}^{\theta} f(a, t) - \sum_{a \in \varrho(v;A)} \sum_{t=0}^{\theta - \tau_i(a)} f(a, t),$$

where $\delta(v; A)$ (resp., $\varrho(v; A)$) represents the set of arcs in A leaving (resp., entering) v. A vector d in $\mathbb{R}^{S \cup T}$ is called an *allocation*, if $d(v) \geq 0$ for every vertex v in S and $d(v) \leq 0$ for every vertex v in T. For each integer i in $\{1, 2\}$, each allocation d in $\mathbb{R}^{S \cup T}$, and each vector w in $\mathbb{R}_+^A$, d is said to be (i, w)-*feasible*, if there exists a function $f : A \times [\Theta] \to \mathbb{R}_+$ satisfying the following conditions.

(D1) Let a and θ be an arc in A and an integer in $[\Theta]$, respectively.
   - If $\theta \leq \Theta - \tau_i(a)$, then $f(a, \theta) \leq q_i \cdot w(a)$.
   - If $\theta > \Theta - \tau_i(a)$, then $f(a, \theta) = 0$.
(D2) Let v and θ be a vertex in V and an integer in $[\Theta]$, respectively.
   - If $v \in V \setminus (S \cup T)$, then $\partial_i f(v, \theta) \leq 0$.
   - If $v \in S$ (resp., T), then $\partial_i f(v, \theta) \leq d(v)$ (resp., $\partial_i f(v, \theta) \geq d(v)$).
(D3) Let v be a vertex in V.
   - If $v \in V \setminus (S \cup T)$, then $\partial_i f(v, \Theta) = 0$.
   - If $v \in S$, then $\partial_i f(v, \Theta) = d(v)$.

For each integer i in $\{1, 2\}$ and each vector w in $\mathbb{R}_+^A$, let $\mathcal{F}_i(w)$ be the set of (i, w)-feasible allocations in $\mathbb{R}^{S \cup T}$. An *assignment* is a tuple (d_1, d_2, w_1, w_2) such that d_1, d_2 are allocations in $\mathbb{R}^{S \cup T}$ and $w_1, w_2 \in \mathbb{R}_+^A$. Furthermore, an assignment (d_1, d_2, w_1, w_2) is said to be *feasible*, if it satisfies the following conditions.

(F1) For every vertex v in S (resp., T), $d_1(v) + d_2(v) = b(v)$ (resp., $\geq u(v)$).
(F2) For every arc a in A, we have $w_1(a) + w_2(a) \leq c(a)$.
(F3) We have $d_1 \in \mathcal{F}_1(w_1)$ and $d_2 \in \mathcal{F}_2(w_2)$.

The goal of MIXED EVACUATION (ME for short) is to decide whether there exists a feasible assignment. Notice that we can straightforwardly formulate ME as a linear programming problem (in Sect. 5, we use an algorithm based on the linear programming). However, since the input size of Θ is $\log_2 \Theta$, its size is not bounded by a polynomial in the input size of ME.

3   Mixed Evacuation with Few Sources and Sinks

In this section, we prove that if $|S \cup T| \le C \log_2 n$ for some constant C, then ME can be solved in polynomial time. Assume that we are given an integer i in $\{1, 2\}$, a vector w in $\mathbb{R}_+^A$, and a subset X of $S \cup T$. Define $\mathbf{D}_i^w(X)$ as the set of functions $f \colon A \times [\Theta] \to \mathbb{R}_+$ satisfying (D1) and the following conditions.

(D4) Let v and θ be a vertex in V and an integer in $[\Theta]$, respectively.
    - If $v \in V \setminus X$, then $\partial_i f(v, \theta) \le 0$.
    - If $v \in X$, then $\partial_i f(v, \theta) \ge 0$.
(D5) For every vertex v in $V \setminus (S \cup T)$, we have $\partial_i f(v, \Theta) = 0$.

Recall that no arc in A enters (resp., leaves) a vertex in S (resp., T). Thus, for every function f in $\mathbf{D}_i^w(X)$ and every vertex v in $(S \setminus X) \cup (T \cap X)$, we have $\partial_i f(v, \Theta) = 0$. Furthermore, we define a function $o_i^w \colon 2^{S \cup T} \to \mathbb{R}_+$ by

$$o_i^w(X) := \max \Big\{ \sum_{v \in X} \partial_i f(v, \Theta) \;\Big|\; f \in \mathbf{D}_i^w(X) \Big\}.$$

Theorem 1 (Klinz [4, Theorem 5.1]). *Assume that we are given an integer i in $\{1, 2\}$, an allocation d in $\mathbb{R}^{S \cup T}$, and a vector w in $\mathbb{R}_+^A$. Then, $d \in \mathcal{F}_i(w)$ if and only if $d(X) \le o_i^w(X)$ for every subset X of $S \cup T$.*

For each pair of vertices s in S and t in T, we denote by $a(t, s)$ an arc from t to s. Define $E := A \cup \{a(t, s) \mid s \in S, t \in T\}$. Furthermore, we define H as the directed graph with the vertex set V and the arc set E. Then, for each integer i in $\{1, 2\}$, each vector w in $\mathbb{R}_+^A$, each subset X of $S \cup T$, and each vector ξ in $\mathbb{R}_+^E$, ξ is called a *feasible static flow* with respect to i, w, and X, if it satisfies the following conditions (S1), (S2), and (S3).

(S1) For every arc a in A, we have $\xi(a) \le q_i \cdot w(a)$.
(S2) For every pair of vertices s in S and t in T, if at least one of $s \in S \setminus X$ and $t \in X$ holds, then $\xi(a(t, s)) = 0$.
(S3) For every vertex v in V, we have $\xi(\delta(v; E)) = \xi(\varrho(v; E))$.

For each integer i in $\{1, 2\}$, each vector w in $\mathbb{R}_+^A$, and each subset X of $S \cup T$, we denote by $\mathbf{S}_i^w(X)$ the set of feasible static flows with respect to i, w, and X. In addition, for each integer i in $\{1, 2\}$, we define a vector k_i in $\mathbb{R}^E$ as follows. For each arc a in $E \setminus A$ (resp., A), we define $k_i(a) := \Theta + 1$ (resp., $-\tau_i(a)$).

Theorem 2 (Ford and Fulkerson [2,3]). *For every integer i in $\{1, 2\}$, every vector w in $\mathbb{R}_+^A$, and every subset X of $S \cup T$, $o_i^w(X)$ is equal to the optimal objective value of the problem of maximizing $\langle k_i, \xi \rangle$ such that $\xi \in \mathbf{S}_i^w(X)$.*

Define $\mathcal{P}$ as the set of assignments (d_1, d_2, w_1, w_2) such that it satisfies (F2) and (F3), and the following condition.

(F1$^+$) For every vertex v in S (resp., T), $d_1(v) + d_2(v) \le b(v)$ (resp., $\ge u(v)$).

Let P1 be the problem of maximizing $d_1(S)+d_2(S)$ such that $(d_1, d_2, w_1, w_2) \in \mathcal{P}$. If the optimal objective value of P1 is equal to $b(S)$, then we can conclude that there exists a feasible assignment. Otherwise, we can conclude that there exists no feasible assignment. This observation implies that if we can formulate P1 by a linear programming problem whose size is bounded by a polynomial in the size of ME, then the polynomial-time solvability of ME follows from the polynomial-time solvability of the linear programming problem [5]. Theorems 1, 2 imply that P1 can be formulated as follows. Define a vector b° in $\mathbb{R}^{S \cup T}$ by $b^\circ(v) := b(v)$ for each vertex v in S and $b^\circ(v) := 0$ for each vertex v in T. Define a vector u° in $\mathbb{R}^{S \cup T}$ by $u^\circ(v) := 0$ for each vertex v in S and $u^\circ(v) := u(v)$ for each vertex v in T.

$$
\begin{aligned}
\text{Maximize} \quad & d_1(S) + d_2(S) \\
\text{subject to} \quad & d_1(v) \geq 0, \; d_2(v) \geq 0 \quad (v \in S) \\
& d_1(v) \leq 0, \; d_2(v) \leq 0 \quad (v \in T) \\
& u^\circ(v) \leq d_1(v) + d_2(v) \leq b^\circ(v) \quad (v \in S \cup T) \\
& \langle k_i, \xi_{i,X} \rangle \geq d_i(X) \quad (i \in \{1,2\}, X \subseteq S \cup T) \\
& \xi_{i,X} \in \mathbf{S}_i^{w_i}(X) \quad (i \in \{1,2\}, X \subseteq S \cup T) \\
& w_1(a) + w_2(a) \leq c(a) \quad (a \in A) \\
& d_1, d_2 \in \mathbb{R}^{S \cup T}, \; w_1, w_2 \in \mathbb{R}_+^A.
\end{aligned}
$$

If $|S \cup T| \leq C \log_2 n$ for some constant C, then it is not difficult to see that the size of this linear programming problem is bounded by a polynomial of the input size of ME. This completes the proof.

4   Mixed Evacuation with Integer Constraints

4.1   Integral Arc Capacities, Supplies, and Sink Capacities

Here we consider INTEGRAL MIXED EVACUATION (IME for short). This problem is a variant of ME in which a feasible assignment (d_1, d_2, w_1, w_2) must satisfy that $d_1, d_2 \in \mathbb{Z}^{S \cup T}$ and $w_1, w_2 \in \mathbb{Z}_+^A$. We prove the **NP**-completeness of IME by reduction from DISJOINT PATHS WITH DIFFERENT COSTS [6] (DPDC for short) defined as follows. In what follows, we do not distinguish a simple directed path in a directed graph and the set of arcs contained in this directed path. We are given a directed graph $G = (N, L)$, a source vertex v^+ in N, and a sink vertex v^- in N. Furthermore, we are given cost vectors ℓ_1, ℓ_2 in $\mathbb{Z}_+^L$ and a non-negative integer h in $\mathbb{Z}_+$. The goal of DPDC is to decide whether there exist arc-disjoint simple directed paths P_1, P_2 from v^+ to v^- such that $\ell_1(P_1) + \ell_2(P_2) \leq h$.

Theorem 3 (Li, McCormick, and Simchi-Levi [6]). *The problem* DPDC *is* **NP**-*complete even if* $h = 0$ *and* $\ell_i(a) \in \{0,1\}$ *for every integer* i *in* $\{1,2\}$ *and every arc* a *in* L.[1]

---

[1] In [6, Theorem 1], although the condition that $h = 0$ and $\ell_i(a) \in \{0,1\}$ for every integer i in $\{1,2\}$ and every arc a in L is not explicitly stated, the reduction in their proof satisfies this condition.

For proving the fact that IME is in **NP**, we need the following theorems. For each finite set U and each function $g\colon 2^U \to \mathbb{R}$, g is said to be *submodular*, if $g(X) + g(Y) \geq g(X \cup Y) + g(X \cap Y)$ for every pair of subsets X, Y of U.

Theorem 4 (E.g., [7,8]). *Assume that we are given a finite set U and a submodular function $g\colon 2^U \to \mathbb{R}$. Then, we can find a subset X of U minimizing $g(X)$ among all subsets of U in time bounded by a polynomial in $|U|$ and* EO, *where* EO *is the time required to compute $g(X)$ for a subset X of U.*

Theorem 5 (Hoppe and Tardos [4]). *For every integer i in $\{1, 2\}$ and every vector w in $\mathbb{R}_+^A$, the function o_i^w is a submodular function.*

Theorem 6 (E.g., [9]). *For every integer i in $\{1, 2\}$ and every subset X of $S \cup T$, we can compute $o_i^{w_i}(X)$ in polynomial time.*

Assume that we are given an integer i in $\{1, 2\}$, an allocation d in $\mathbb{R}^{S \cup T}$, and a vector w in $\mathbb{R}_+^A$. Then, Theorem 1 implies that $d \in \mathcal{F}_i(w)$ if and only if $o_i^w(X) - d(X) \geq 0$ for a subset X of $S \cup T$ minimizing $o_i^w(X) - d(X)$ among all subsets of $S \cup T$. Thus, since Theorem 5 implies that $o_i^w - d$ is submodular, Theorems 4, 6 imply that we can check whether $d \in \mathcal{F}_i(w)$ in polynomial time.[2]

Theorem 7. *The problem* IME *is* **NP**-*complete.*

Proof. Theorems 1, 4, 5, and 6 imply that IME is in **NP**. We prove that IME is **NP**-complete by reduction from DPDC. Assume that we are given an instance of DPDC such that $h = 0$ and $\ell_i(a) \in \{0, 1\}$ for every integer i in $\{1, 2\}$ and every arc a in L. Then, we construct an instance of IME as follows. Define $V := N \cup \{s^*\}$, where s^* is a new vertex. Define $A := L \cup \{a_1, a_2\}$, where a_1 and a_2 are new arcs from s^* to v^+. Define $S := \{s^*\}$ and $T := \{v^-\}$. Define $c(a) := 1$ for each arc a in A. For each arc a in A, we define

$$
\tau_1(a) := \begin{cases} \ell_1(a) & \text{if } a \in L \\ 0 & \text{if } a = a_1 \\ 1 & \text{if } a = a_2 \end{cases}
\qquad
\tau_2(a) := \begin{cases} \ell_2(a) & \text{if } a \in L \\ 1 & \text{if } a = a_1 \\ 0 & \text{if } a = a_2. \end{cases}
$$

Define $b(s^*) := 2$ and $u(v^-) := -2$. Define $q_1 := 1$, $q_2 := 1$, and $\Theta := 0$.

Assume that there are arc-disjoint simple directed paths P_1, P_2 in G from v^+ to v^- such that $\ell_1(P_1) + \ell_2(P_2) \leq 0$. Since $\ell_i(a) \geq 0$ for every integer i in $\{1, 2\}$ and every arc a in L, we have $\ell_1(P_1) = \ell_2(P_2) = 0$. For each integer i in $\{1, 2\}$, we define a directed path P_i^+ in D as the directed path obtained by adding a_i to P_i. Since P_1 and P_2 are arc-disjoint, P_1^+ and P_2^+ are arc-disjoint. Furthermore, for every integer i in $\{1, 2\}$, since $\ell_i(P_i) = 0$ and $\tau_i(a_i) = 0$, we have $\tau_i(P_i^+) = 0$. For each integer i in $\{1, 2\}$, we define $d_i(s^*) := 1$ and $d_i(v^-) := -1$. In addition, for each integer i in $\{1, 2\}$, we define a vector w_i in $\mathbb{Z}_+^A$ as follows. If $a \in P_i^+$, then we define $w_i(a) := 1$. Otherwise, we define $w_i(a) := 0$. Since P_1^+ and P_2^+ are

---

[2] This proof is the same as the proof of the polynomial-time solvability of the decision version of the quickest transshipment problem in [4].

arc-disjoint, $w_1(a) + w_2(a) \le 1 = c(a)$ for every arc a in A. Thus, (d_1, d_2, w_1, w_2) is a feasible assignment.

Next we assume that there exists a feasible assignment (d_1, d_2, w_1, w_2) such that $d_1, d_2 \in \mathbb{Z}^{S \cup T}$ and $w_1, w_2 \in \mathbb{Z}_+^A$. Since $\tau_1(a_2) = \tau_2(a_1) = 1$, $c(a_1) = c(a_2) = 1$, and $\Theta = 0$, we have $d_1(s^*) = d_2(s^*) = 1$. Since $c(a) = 1$ for every arc a in A, we have $w_1(a), w_2(a) \in \{0, 1\}$ and at most one of $w_1(a)$ and $w_2(a)$ is equal to 1 for every arc a in A. For each integer i in $\{1, 2\}$, we denote by L_i the sets of arcs a in A such that $w_i(a) = 1$. Then, L_1 and L_2 are disjoint. For every integer i in $\{1, 2\}$, since $d_i(s^*) = 1$ and $\Theta = 0$, L_i contains a simple directed path L_i' from s^* to v^- such that $\tau_i(L_i') = 0$ as a subset. Furthermore, for every integer i in $\{1, 2\}$, the definition of τ_i, we have $a_i \in L_i'$. For each integer i in $\{1, 2\}$, let P_i be the directed path obtained by removing a_i from L_i'. Then, for every integer i in $\{1, 2\}$, we have $\ell_i(P_i) = 0$. This completes the proof.   □

4.2   Integral Supplies and Sink Capacities

Here we consider problems of finding an integral allocation of supplies and sink capacities. We consider INTEGRAL MIXED EVACUATION WITH ARC CAPACITIES (IMEAC for short) defined as follows. We are given vectors $w_1, w_2 \in \mathbb{Z}_+^A$ such that $w_1(a) + w_2(a) \le c(a)$ for every arc a in A. Then, the goal is to decide whether there exists a feasible assignment (d_1, d_2, w_1, w_2) such that $d_1, d_2 \in \mathbb{Z}^{S \cup T}$. We prove that IMEAC can be solved in polynomial time. In the rest of this section, we define $o_i := o_i^{w_i}$ for each integer i in $\{1, 2\}$. For each finite set U, each function $g: 2^U \to \mathbb{R}$, and each pair of subsets P_1, P_2 of $\mathbb{R}^U$, we define $\mathrm{P}(g) := \{x \in \mathbb{R}^U \mid x(X) \le g(X) \ (\forall X \subseteq U)\}$ and $P_1 + P_2 := \{x + y \mid x \in P_1, y \in P_2\}$. Then, Theorem 1 implies that IMEAC can be formulated as the following problem P2.

$$
\begin{array}{ll}
\text{Maximize} & d_1(S) + d_2(S) \\
\text{subject to} & d_1(v) \ge 0, \ d_2(v) \ge 0 \quad (v \in S) \\
& d_1(v) \le 0, \ d_2(v) \le 0 \quad (v \in T) \\
& u^\circ(v) \le d_1(v) + d_2(v) \le b^\circ(v) \quad (v \in S \cup T) \\
& d_1 \in \mathrm{P}(o_1), \ d_2 \in \mathrm{P}(o_2), \ d_1, d_2 \in \mathbb{Z}^{S \cup T}.
\end{array}
$$

If the optimal objective value is equal to $b(S)$, then we can conclude that there exists a desired assignment. Otherwise, there exists no such an assignment.

Assume that we are given an integer i in $\{1, 2\}$. Define $o_i^- : 2^{S \cup T} \to \mathbb{R}_+$ by setting $o_i^-(X)$ to be the minimum value of $o_i(Y)$ over all subsets Y of X such that $X \cap S \subseteq Y$. Let X be a subset of $S \cup T$. Define a function $o_{i,X} : 2^{X \setminus S} \to \mathbb{R}$ by setting $o_{i,X}(Y) := o_i(Y \cup (X \cap S)) - o_i(X \cap S)$. It is not difficult to see that $o_{i,X}$ is submodular and $o_i^-(X) = \min\{o_{i,X}(Y) \mid Y \subseteq X \setminus S\} + o_i(X \cap S)$. That is, we can evaluate $o_i^-(X)$ by evaluating the value of o_i and using the algorithm for submodular function minimization. It is known [10, Eq. (3.10)] that o_i^- is a submodular function. Furthermore, it is known [10, Theorem 3.3] that $\mathrm{P}(o_i^-)$ is equal to the set of vectors d in $\mathrm{P}(o_i)$ such that $d(v) \le 0$ for every vertex v in T.

Thus, P2 is equivalent to the following problem.

$$
\begin{array}{ll}
\text{Maximize} & d_1(S) + d_2(S) \\
\text{subject to} & d_1(v) \geq 0,\ d_2(v) \geq 0 \quad (v \in S) \\
& u^\circ(v) \leq d_1(v) + d_2(v) \leq b^\circ(v) \quad (v \in S \cup T) \\
& d_1 \in \mathrm{P}(o_1^-),\ d_2 \in \mathrm{P}(o_2^-),\ d_1, d_2 \in \mathbb{Z}^{S \cup T}.
\end{array}
$$

We consider the following problem P3.

$$
\begin{array}{ll}
\text{Maximize} & d_1(S) + d_2(S) \\
\text{subject to} & u^\circ(v) \leq d_1(v) + d_2(v) \leq b^\circ(v) \quad (v \in S \cup T) \\
& d_1 \in \mathrm{P}(o_1^-),\ d_2 \in \mathrm{P}(o_2^-),\ d_1, d_2 \in \mathbb{Z}^{S \cup T}.
\end{array}
$$

Lemma 1. *The optimal objective values of* P2 *and* P3 *are the same.*

Proof. For each optimal solution (d_1, d_2) of P3, we define $\gamma(d_1, d_2)$ as the number of pairs (i, v) of an integer i in $\{1, 2\}$ and a vertex v in S such that $d_i(v) < 0$. Let (d_1, d_2) be an optimal solution of P3 minimizing $\gamma(d_1, d_2)$ among all optimal solutions of P3. If $\gamma(d_1, d_2) = 0$, then since P3 is a relaxation problem of P2, (d_1, d_2) is clearly an optimal solution of P2, and thus the proof is done. Assume that $\gamma(d_1, d_2) \geq 1$. Let (i, v) be a pair of an integer i in $\{1, 2\}$ and a vertex v in S such that $d_i(v) < 0$. We assume that $i = 1$ (we can treat the case of $i = 2$ in the same way). For proving this lemma by contradiction, we prove that there exists an optimal solution (d_1', d_2') of P3 such that $\gamma(d_1, d_2) > \gamma(d_1', d_2')$. This contradicts the definition of (d_1, d_2), and thus this completes the proof.

Define vectors d_1', d_2' in $\mathbb{Z}^{S \cup T}$ as follows. Define $d_1'(v') := d_1(v')$ and $d_2'(v') := d_2(v')$ for each vertex v' in $(S \cup T) \setminus \{v\}$. Furthermore, define $d_1'(v) := 0$ and $d_2'(v) := \min\{d_2(v), b(v)\}$. We first prove that $d_1(v) + d_2(v) \leq d_1'(v) + d_2'(v)$. If $d_2'(v) = d_2(v)$, then since $d_1(v) < 0$, this clearly holds. If $d_2'(v) = b(v)$, then since $d_1(v) + d_2(v) \leq b(v)$, this clearly holds. This implies that the objective value of (d_1', d_2') is no less than that of (d_1, d_2). Thus, what remains is to prove that (d_1', d_2') is a feasible solution of P3. The above inequality implies that (d_1', d_2') satisfies the first constraint of P3. In addition, d_2' clearly belongs to $\mathrm{P}(o_2^-)$. Thus, it suffices to prove that $d_1' \in \mathrm{P}(o_1^-)$. Assume that this does not hold. Then, there exists a subset of X of $S \cup T$ such that $v \in X$ and $o_1^-(X) - d_1(X) < -d_1(v)$. Since it is not difficult to see that $o_1(Y \setminus \{v\}) \leq o_1(Y)$ for every subset Y of X such that $X \cap S \subseteq Y$, $o_1^-(X \setminus \{v\}) \leq o_1^-(X)$. Thus, $o_1^-(X \setminus \{v\}) < d_1(X \setminus \{v\})$, which contradicts that $d_1 \in \mathrm{P}(o_1^-)$. This completes the proof. □

Lemma 2. *For every integer i in $\{1, 2\}$ and every subset X of $S \cup T$, we have $o_i(X) \in \mathbb{Z}$, which implies that $o_i^-(X) \in \mathbb{Z}$.*

Proof. This lemma follows from Theorem 2 and [11, Theorem 12.8]. □

Theorem 8 (E.g., [11, Corollary 46.2c]**).** *Assume that we are given a finite set U and submodular functions $\sigma, \pi \colon 2^U \to \mathbb{Z}$ such that $\sigma(\emptyset) = \pi(\emptyset) = 0$. Then, we have $(\mathrm{P}(\sigma) \cap \mathbb{Z}^U) + (\mathrm{P}(\pi) \cap \mathbb{Z}^U) = (\mathrm{P}(\sigma) + \mathrm{P}(\pi)) \cap \mathbb{Z}^U$.*[3]

Lemmas 1, 2 and Theorem 8 imply that P2 is equivalent to the following problem.

$$
\begin{aligned}
\text{Maximize} \quad & d(S) \\
\text{subject to} \quad & u^\circ(v) \leq d(v) \leq b^\circ(v) \quad (v \in S \cup T) \\
& d \in \mathrm{P}(o_1^-) + \mathrm{P}(o_2^-), \ d \in \mathbb{Z}^{S \cup T}.
\end{aligned}
$$

Theorem 9 (E.g., [11, Theorem 44.6]**).** *Assume that we are given a finite set U and submodular functions $\sigma, \pi \colon 2^U \to \mathbb{Z}$ such that $\sigma(\emptyset) = \pi(\emptyset) = 0$. Then, we have $\mathrm{P}(\sigma) + \mathrm{P}(\pi) = \mathrm{P}(\sigma + \pi)$.*[4]

Theorem 9 implies that P2 is equivalent to the following problem.

$$
\begin{aligned}
\text{Maximize} \quad & d(S) \\
\text{subject to} \quad & u^\circ(v) \leq d(v) \leq b^\circ(v) \quad (v \in S \cup T) \\
& d \in \mathrm{P}(o_1^- + o_2^-), \ d \in \mathbb{Z}^{S \cup T}.
\end{aligned}
$$

We consider the following relaxation problem LP2 of P2.

$$
\begin{aligned}
\text{Maximize} \quad & d(S) \\
\text{subject to} \quad & u^\circ(v) \leq d(v) \leq b^\circ(v) \quad (v \in S \cup T) \\
& d \in \mathrm{P}(o_1^- + o_2^-).
\end{aligned}
$$

Lemma 3. *The optimal objective values of P2 and LP2 are the same.*

Proof. Since o_1^-, o_2^- are submodular functions, $o_1^- + o_2^-$ is a submodular function. Furthermore, Lemma 2 implies that $o_1^-(X) + o_2^-(X) \in \mathbb{Z}$ for every subset X of $S \cup T$. Thus, this lemma follows from [11, Corollary 44.3c] (i.e., the box-total dual integrality of the constraints corresponding to $\mathrm{P}(o_1^- + o_2^-)$). □

Theorem 10. *The problem IMEAC can be solved in polynomial time.*

Proof. In the same way as the algorithm described after Theorem 6, we can check in polynomial time whether $d \in \mathrm{P}(o_1^- + o_2^-)$ for a given vector d in $\mathbb{R}^{S \cup T}$ by minimizing the submodular function $o_1^- + o_2^- - d$. In addition, we can check in polynomial time whether a given vector d in $\mathbb{R}^{S \cup T}$ satisfies the first constraint of LP2. Thus, we can solve the separation problem for LP2 by using Theorem 4 in polynomial time (if $d \notin \mathrm{P}(o_1^- + o_2^-)$, then a separating hyperplane can be obtained from a minimizer of $o_1^- + o_2^- - d$). This implies that we can solve IMEAC in polynomial time by using the results of [12] (see also [13, Theorem 6.36]). □

---

[3] Precisely speaking, [11, Corollary 46.2c] considers $\mathrm{P}(\sigma) \cap \mathbb{R}_+^U$. However, the similar result holds for $\mathrm{P}(\sigma)$ (see the paragraph after the proof of [11, Theorem 44.6]).

[4] In [11, Theorem 44.6], the monotonicity of functions are assumed. However, even if functions are not monotone, this theorem holds. See also [10, Eq. (3.32)].

4.3 Unsplittable Supplies and Sink Capacities

Here we consider the following variant of IMEAC called UNSPLITTABLE MIXED EVACUATION WITH ARC CAPACITIES (UMEAC for short). In this problem, we are given vectors $w_1, w_2 \in \mathbb{Z}_+^A$ such that $w_1(a) + w_2(a) \leq c(a)$ for every arc a in A. The goal is to decide whether there exists a feasible assignment (d_1, d_2, w_1, w_2) such that $d_1(v), d_2(v) \in \{0, b(v)\}$ for every vertex v in S and $d_1(v), d_2(v) \in \{0, u(v)\}$ for every vertex v in T. In what follows, we prove that UMEAC is **NP**-complete. Notice that if $|S \cup T| \leq C \log_2 n$ for some constant C, then it follows from Theorems 1, 4, 5, and 6 that UMEAC can be solved in polynomial time by enumerating all subsets of $S \cup T$. We will prove the **NP**-completeness of UMEAC by reduction from PARTITION. In this problem, we are given a finite set I and a vector π in $\mathbb{Z}_+^I$ such that $\pi(I)$ is even. Then, the goal is to decide whether there exists a subset J of I such that $\pi(J) = \pi(I \setminus J)$. It is well known [14] that PARTITION is **NP**-complete.

Theorem 11. *The problem* UMEAC *is* **NP***-complete.*

Proof. In the same ways as the proof of Theorem 7, we can prove that UMEAC is in **NP**. We prove that the **NP**-completeness of UMEAC by reduction from PARTITION. Assume that we are given an instance of PARTITION, and then we construct an instance of UMEAC as follows. Define $V := \{v_i \mid i \in I\} \cup \{v^\circ, v^\bullet, v_1^*, v_2^*\}$ and $A := \{(v_i, v^\circ), (v_i, v^\bullet) \mid i \in I\} \cup \{(v^\circ, v_1^*), (v^\bullet, v_2^*)\}$. Define $S := \{v_i \mid i \in I\}$ and $T := \{v_1^*, v_2^*\}$. Define $c(a) := 1$, $\tau_1(a) := 0$, and $\tau_2(a) := 0$ for each arc a in A. Define $b(v_i) := \pi(i)$ for each element i in I. Define $u(v_1^*) := -\pi(I)$ and $u(v_2^*) := -\pi(I)$. Define $q_1 := 1$ and $q_2 := 1$. Define $\Theta := (\pi(I)/2) - 1$. Lastly, we define vectors w_1, w_2 in $\mathbb{Z}_+^A$ as follows. For each arc $a = (x, y)$ in A, we define

$$w_1(a) := \begin{cases} 1 & \text{if } x = v^\circ \text{ or } y = v^\circ \\ 0 & \text{otherwise} \end{cases} \qquad w_2(a) := \begin{cases} 1 & \text{if } x = v^\bullet \text{ or } y = v^\bullet \\ 0 & \text{otherwise.} \end{cases}$$

Assume that there exists a subset J of I such that $\pi(J) = \pi(I \setminus J)$. Then, we define vectors d_1, d_2 in $\mathbb{Z}^{S \cup T}$ by

$$d_1(v) \text{ (resp., } d_2(v)) := \begin{cases} \pi(i) \text{ (resp., } 0) & \text{if } v = v_i \text{ for some } i \in J \\ 0 \text{ (resp., } \pi(i)) & \text{if } v = v_i \text{ for some } i \in I \setminus J \\ -\pi(I) \text{ (resp., } 0) & \text{if } v = v_1^* \\ 0 \text{ (resp., } -\pi(I)) & \text{if } v = v_2^*. \end{cases}$$

Since $\pi(J) = \pi(I \setminus J) = \pi(I)/2$, (d_1, d_2, w_1, w_2) is a feasible assignment.

Conversely, we assume that there exists a feasible assignment (d_1, d_2, w_1, w_2) such that $d_1(v), d_2(v) \in \{0, b(v)\}$ for every vertex v in S and $d_1(v), d_2(v) \in \{0, u(v)\}$ for every vertex v in T. Since $\Theta = (\pi(I)/2) - 1$, we have $d_1(S) = \pi(I)/2$ and $d_2(S) = \pi(I)/2$. Thus, if we define J as the set of elements i in I such that $d_1(v_i) = b(v_i)$, then $\pi(J) = \pi(I \setminus J)$. This completes the proof. □

5   Case Study

Here we apply our model to the case study of Minabe town in Wakayama prefecture, which was designated as those in which safe evacuation from tsunami is difficult when Nankai Trough Earthquake occurs. The population of Minabe town is about 12000. According to the census data of 2013, the number of people living in the tsunami inundation area of this town is 4745. The left figure of Fig. 1 shows the map of this area and the expected height of the tsunami caused by Nankai Trough Earthquake. The town is surrounded by mountains of height ranging from 100 to 200 m.
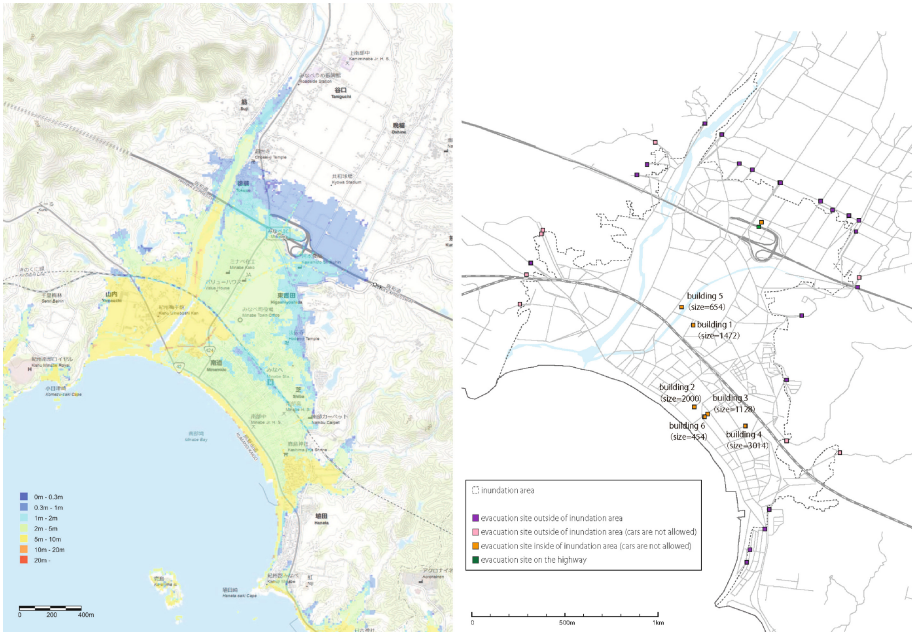

Fig. 1. (Left) The target area and its inundation depth. (Right) The road network in the target areas and evacuation sites.

It is predicted that in twelve minutes after that earthquake occurs, the first tsunami of height 1 m arrives, and then that of height 5m (and of 10 m, respectively) will arrive after 15 min (and 24 min, respectively). Since people usually start evacuation five minutes after the earthquake occurs, the actual time remaining for evacuation is from five to fifteen minutes depending on where they live. Since there are not enough evacuation buildings in the center of the town, most of the people will have to go to the outside of the tsunami inundation area, and thus some of them may not succeed to evacuate to a safety place.

Under this circumstance, we consider the following experiments. Our computational experiment aims at the inundation area of Minabe town whose

population is 4745. We prepare two scenarios. The first one is that people should have to evacuate to the outside of the inundation area. The second is that people should have to evacuate to the outside of the inundation area or to tsunami evacuation buildings located inside of the inundation area. There exist six evacuation buildings inside the inundation area (numbered from 1 through 6 in the right figure of Fig. 1) whose sizes (i.e., the maximum number of evacuees that can be accommodated) are 1472, 2000, 1128, 3014, 654 and 454, respectively. We constructed a model of a dynamic network by using the GIS databases: the fundamental map information (1/2500, the Geospatial Information Authority of Japan), the population census (2010, the Ministry of Internal Affairs and Communications of Japan), and the Japan digital road map (Japan Digital Road Map Association). The road network has 860 nodes and 1,106 arcs.

We assign to a sink vertex the capacity of the evacuation site located at the vertex, i.e., the maximum number of evacuees that the site accommodates. In our experiment, the capacity of a building was computed based on the available floor space, assuming that two persons per m^2 can be accommodated. The capacity of an evacuation site which is outside the tsunami inundation area is assumed to be infinity. However, since a hill top may have an upper limit on the number of evacuees that can be accommodated, its capacity is estimated based on an aerial photograph. Evacuation by cars is only possible to the outside of the tsunami inundation area, and thus is assumed to be not allowed to tsunami evacuation buildings or hill tops. Since there are not enough tsunami evacuation buildings, the delay of evacuation is predicted. (In our experiment, we solve ME by using a linear programming solver. Thus, we can add additional constraints to our model. Furthermore, the minimum evacuation completion time can be computed by the binary search.)

5.1   Computational Results

We use Gurobi Optimizer (see http://www.gurobi.com/) as the solver to solve linear programs corresponding to our experimental data.

As seen from Table 1, in each scenario, the result for the case where cars are allowed to use is much better in the minimum evacuation completion time than the one where they are not allowed. Comparing the scenario 2 with the scenario 1, the number of evacuees who walked to the evacuation site increased since evacuation buildings located in the town center can be used in the scenario 2.

Table 1. Computational results of each scenario.

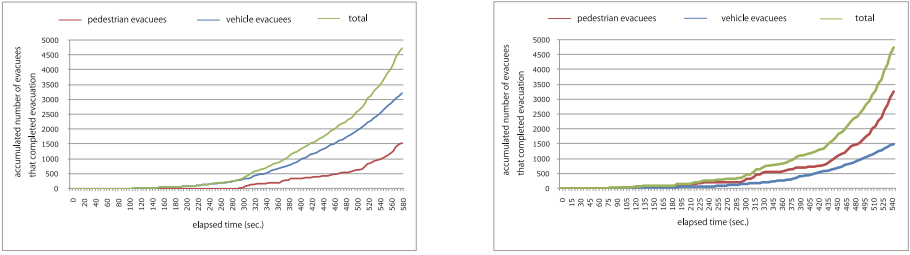| Scenario	Evacuation time	Percentage of car usage	Pedestrians only
1	9m40s	68.1 %	18m00s
2	9m05s	31.4 %	17m30s

Fig. 2. (Left) The transition of the accumulated number of evacuees that completed evacuation in the scenario 1. (Right) The transition of the accumulated number of evacuees that completed evacuation in the scenario 2.

Now let us look at Fig. 2 that shows how the number of evacuees that have completed evacuation increases as time proceeds since the evacuation starts. It is observed that in the latter half for the whole time period, the number of evacuees that completed evacuation rapidly increases in both scenarios. Ideally, it is desired that the number of evacuees that completed evacuation is large in the early stage. This point should be taken into account in order to improve the current model.

Fig. 3. (Left) Distribution of evacuees that used cars in the scenario 1. (Right) Distribution of evacuees that used cars in the scenario 2. (Color figure online)
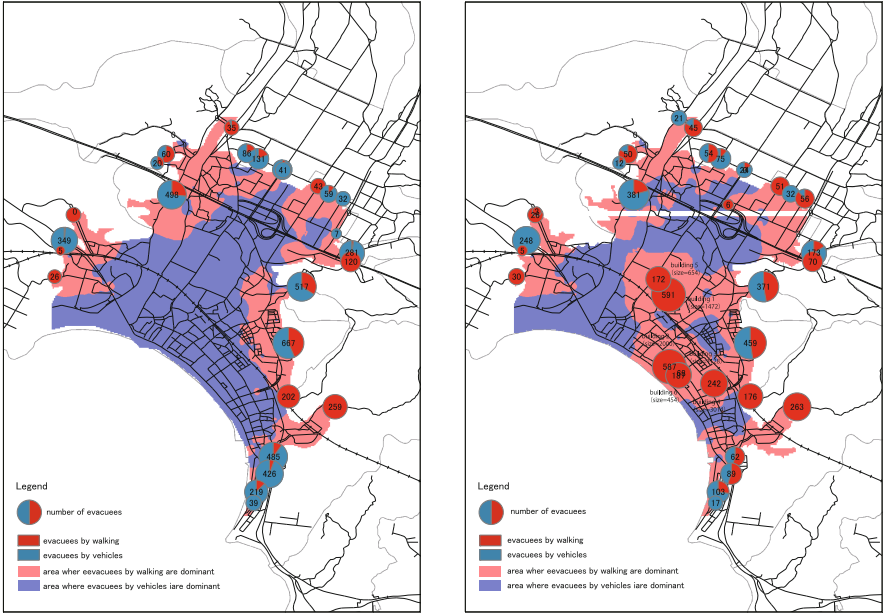
Fig. 4. (Left) The number of evacuees that arrived at each evacuation site, and the ratio of evacuees who arrived at the site by walking and those who arrived there by a car in the scenario 1 (Right) The number of evacuees that arrived at each evacuation site, and the ratio of evacuees who arrived at the site by walking and those who arrived there by a car in the scenario 2.

Let us look at the way of evacuation (by walking or a car) at each vertex. In Fig. 3, if the color at each vertex is close to blue, it means that a majority of people used cars for evacuation while on the other hand, if it is close to red, a majority of people walked for evacuation. Comparing the scenarios 1 and 2, the car usage significantly decreased in the scenario 2 near the coast since there are evacuation buildings nearby.

Figure 4 shows the number of evacuees that arrived at each evacuation site, and the ratio of evacuees who arrived at the site by walking and those who arrived there by a car. In the scenario 1, for most of evacuation sites, the number of evacuees who arrived by cars exceeds that of evacuees who arrived by walking. On the other hand in the scenario 1, many evacuees living near the town center evacuated to evacuation buildings inside the inundation area.

6   Conclusion

In this paper, we introduce the mixed evacuation problem that is motivated by making an evacuation plan in an emergent situation in which people can evacuation on foot or by car. We study this problem from the theoretical and practical viewpoints. An apparent future work from the theoretical viewpoint

is to reveal the computational complexity of the mixed evacuation problem in the general case. From the practical viewpoint, it is a future work to apply our model to areas other than Minabe town. There exist many small towns on the coastal area facing the Pacific Ocean whose local governments are faced with a serious problem that they have to spend a significant percentage of their budget for building a tsunami evacuation buildings in order to reduce the loss of human lives from tsunami triggered by Nankai Trough Earthquake that are expected to occur with 70 % within the coming 30 years [1]. In this respect, we hope that the methods developed for facility location problems will help to reduce the budget to be used for such disaster prevention.

References

1. The Headquarters for Earthquake Research Promotion: Evaluation of Long-Term Probability of Active Fault and Subduction-Zone Earthquake Occurrence (in Japanese) (2016). http://www.jishin.go.jp/main/choukihyoka/ichiran.pdf
2. Ford Jr., L.R., Fulkerson, D.R.: Constructing maximal dynamic flows from staticflows. Oper. Res. **6**(3), 419–433 (1958)
3. Ford Jr., L.R., Fulkerson, D.R.: Flows in Networks. Princeton University Press, Princeton (1962)
4. Hoppe, B., Tardos, É.: The quickest transshipment problem. Math. Oper. Res. **25**(1), 36–62 (2000)
5. Khachiyan, L.G.: Polynomial algorithms in linear programming. USSR Comput. Math. Math. Phys. **20**(1), 53–72 (1980)
6. Li, C., McCormick, S.T., Simchi-Levi, D.: Finding disjoint paths with different path-costs: complexity and algorithms. Networks **22**(7), 653–667 (1992)
7. Schrijver, A.: A combinatorial algorithm minimizing submodular functions in strongly polynomial time. J. Comb. Theor. Ser. B **80**(2), 346–355 (2000)
8. Iwata, S., Fleischer, L., Fujishige, S.: A combinatorial strongly polynomial algorithm for minimizing submodular functions. J. ACM **48**(4), 761–777 (2001)
9. Orlin, J.B.: A faster strongly polynomial minimum cost flow algorithm. Oper. Res. **41**(2), 338–350 (1993)
10. Fujishige, S.: Submodular Functions and Optimization. Annals of Discrete Mathematics, vol. 58, 2nd edn. Elsevier, Amsterdam (2005)
11. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency, vol. 24. Springer, Heidelberg (2003)
12. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization, 2nd edn. Springer, Heidelberg (1993)
13. Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A.: Combinatorial Optimization. Wiley, Hoboken (1998)
14. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H Freeman and Company, New York (1979)