<table>
<thead>
<tr>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>An Efficient Algorithm for Evacuation Problems in Dynamic Network Flows with Uniform Arc Capacity (Algorithmic aspects in information and management)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kamiyama, Naoyuki; Katoh, Naoki; Takizawa, Atsushi</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture notes in computer science (2006), 4041: 231-242</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issue Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://hdl.handle.net/2433/84857">http://hdl.handle.net/2433/84857</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>The original publication is available at <a href="http://www.springerlink.com">www.springerlink.com</a>; This is not the published version. Please cite only the published version.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conference Paper</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Textversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
</tr>
</tbody>
</table>

Kyoto University
An Efficient Algorithm for Evacuation Problems in Dynamic Network Flows with Uniform Arc Capacity

Naoyuki Kamiyama *, Naoki Katoh, and Atsushi Takizawa

Department of Architecture and Architectural Engineering, Kyoto University
Kyotodaigaku-Katsura, Nishikyo-ku, Kyoto, 615-8540, Japan
{is.kamiyama,naoki,kukure}@archi.kyoto-u.ac.jp

Abstract. In this paper, we consider the quickest flow problem in a network which consists of a directed graph with capacities and transit times on its arcs. We present an $O(n \log n)$ time algorithm for the quickest flow problem in a network of grid structure with uniform arc capacity which has a single sink where $n$ is the number of vertices in the network.

1 Introduction

It is very important to establish crisis management systems against large-scale disasters such as big earthquakes, conflagrations and tsunamis. We need to consider the crisis management against disasters to secure evacuation pathways and to effectively guide residents to a safe place. In our work, we adopt dynamic network flows as a model for evacuation. A dynamic network flow is defined on a network which consists of a directed graph $D = (V, A)$ with capacity $c(e)$ and transit time $\tau(e)$ on every arc $e \in A$. For example, if we consider urban evacuation, vertices model buildings, rooms, exits and so on, and an arc models a pathway or a road connecting vertices. For an arc $e$, capacity $c(e)$ represents the number of people which can traverse the arc $e$ per unit time, and $\tau(e)$ denotes the time it takes to traverse $e$. Given a network with initial supplies at vertices, the problem is to find an optimal dynamic network flow such that we can send all the initial supplies to sinks as quickly as possible. In the case where a network has several sources and sinks which have specified supply or demand respectively, this problem can be solved by the algorithm of Hoppes and Tardos [1] in polynomial time. However their running time is high-order polynomial, and hence is not practical in general. So it is necessary to devise a faster algorithm for a tractable and practically useful subclass of this problem.

In this paper, we restrict our attention to grid networks with uniform arc capacity. The condition that arc capacity is uniform is practically acceptable because the width of road or corridor is generally standardized. Restriction of network structure to grid networks is useful since such structure often appears

* Supported by JSPS Grant-in-Aid for Scientific Research on priority areas of New Horizons in Computing
in modelling building corridors and city streets. We present an $O(n \log n)$ time algorithm for the quickest flow problem in a network of grid structure with uniform arc capacity which has a single sink where $n$ is the number of vertices in the network.

**Previous Works.** As mentioned above, Hoppes and Tardos proposed a polynomial time algorithm for the problem [1]. As a special class of networks, Mamada et al. [2] considered tree networks with a single sink and presented an $O(n \log^2 n)$ time algorithm. For the case of tree networks with multiple sinks, Mamada et al. [3] presented an $O(n \log^3 n)$ time algorithm. For the case of tree networks with multiple sinks, Mamada et al. [4] presented an $O(n^2 k \log^2 n)$ time algorithm for $k$-sink case under the restriction that all the supplies going through a common vertex are sent to a single sink. However, to the authors’ knowledge, no one has ever studied special class of networks other than tree networks for the evacuation problem.

**Organization.** Section 2 gives necessary definitions and preliminaries. Section 3 considers the quickest flow problem for grid networks with uniform arc capacity and proposes an $O(n \log n)$ time algorithm. Section 4 concludes the paper.

## 2 Problem Formulation and Notations

We consider a network $\mathcal{N} = (D = (V, A), c, \tau, b_v, V^*)$, where $D$ is a directed graph, $V$ is a set of vertices, $A$ is a set of arcs, $c: A \rightarrow \mathbb{R}^+$ is the upper bound for the rate of flow that enters each arc per unit time, $\tau: A \rightarrow \mathbb{Z}^+$ is a transit time function, $b_v \in \mathbb{R}^+$ gives an initial supply of $v \in V$, and $V^* \subset V$ is a set of sinks. Here $\mathbb{R}^+$ denotes the set of nonnegative reals and $\mathbb{Z}^+$ denotes the set of nonnegative integers. For simplicity, we write $c(v, w)$ and $\tau(v, w)$ instead of $c((v, w))$ and $\tau((v, w))$ respectively for any $(v, w) \in A$. Given a network, our problem is to compute the minimum time required to send all supplies to sinks.

Here we define a discrete-time dynamic network flow $f: A \times \mathbb{Z}_+ \rightarrow \mathbb{R}_+$. For any arc $e \in A$ and $\theta \in \mathbb{Z}_+$, we denote by $f(e, \theta)$ the flow rate entering the arc $e$ at time $\theta$ which arrives at the head of $e$ at time $\theta + \tau(e)$. We call $f: A \times \mathbb{Z}_+ \rightarrow \mathbb{R}_+$ a feasible dynamic flow in $\mathcal{N}$ if it satisfies the following three conditions, i.e., capacity constraint, flow conservation, and demand constraint [2].

**Capacity constraint:** For any arc $e \in A$ and $\theta \in \mathbb{Z}_+$,

$$0 \leq f(e, \theta) \leq c(e). \quad (1)$$

**Flow conservation:** For any $v \in V$ and $\Theta \in \mathbb{Z}_+$,

$$\sum_{e \in \Delta^+(v)} \theta f(e, \theta) - \sum_{e \in \Delta^-(v)} \theta f(e, \theta - \tau(e)) \leq b_v. \quad (2)$$

**Demand constraint:** There exists a time $\Theta \in \mathbb{Z}_+$ such that

$$\sum_{e \in \Delta^-(V^*)} \theta f(e, \theta - \tau(e)) - \sum_{e \in \Delta^+(V^*)} \theta f(e, \theta) = \sum_{v \in V \setminus V^*} b_v. \quad (3)$$
Here $\Delta^+(V') \equiv \{(v, w) \in A \mid v \in V', w \notin V'\}$, and $\Delta^-(V') \equiv \{(v, w) \in A \mid v \notin V', w \in V'\}$ for any $V' \subseteq V$. For simplicity, we write $\Delta^+(v)$ and $\Delta^-(v)$ instead of $\Delta^+(\{v\})$ and $\Delta^-(\{v\})$, respectively. For a feasible dynamic flow $f$, let $\Theta(f)$ denote the completion time for $f$, i.e., the minimum time $\Theta$ satisfying (3), and let $\mathcal{F}_p$ denote the set of all feasible dynamic flows in $\mathcal{N}$. The quickest flow problem asks to find a feasible dynamic flow $f$ that minimizes $\Theta(f)$.

Here we define flow-table [2] which is a function from $\mathbb{Z}_+$ to $\mathbb{R}_+$. There are two kinds of flow-tables, arriving-table $\text{AT}_v$, for each vertex $v \in V$, and sending-table $\text{ST}_e$ for each arc $e \in A$. Arriving-table $\text{AT}_v$ represents the sum of the flow rates arriving at the vertex $v$ as a function of time $\theta$, i.e.,

$$\text{AT}_v(\theta) = B_v(\theta) + \sum_{e \in \Delta^-(v)} f(e, \theta - \tau(e))$$

where we regard the initial supply $b_v$ as a flow-table $B_v$ as follows: $B_v(0) = b_v$ and $B_v(\theta) = 0$ if $\theta \neq 0$. Sending table $\text{ST}_e$ represents the flow rate entering the arc $e$ as a function of time $\theta$, i.e.,

$$\text{ST}_e(\theta) = f(e, \theta).$$

We define $T$ as a function of flow-table $FT$ as follows: $T(FT) = \max\{\theta \in \mathbb{Z}_+ \mid FT(\theta) > 0\}$. $T(\text{ST}_e) + \tau(e)$ represents the time to complete the evacuation from $e$.

In this paper, we will focus our attention on grid graph as an underlying graph of a network. For simplicity, we assume a grid graph is on $N^2$ grid points $\{1, \ldots, N\} \times \{1, \ldots, N\}$ in the plane, and let $n = N^2$. Here a vertex is identified with $(i, j)$ with $1 \leq i \leq N$ and $1 \leq j \leq N$. A sink $r$ is specified as one of vertices. The distance between two vertices $(i, j)$ and $(i', j')$ is defined as $|i - i'| + |j - j'|$. Two vertices $(i, j)$ and $(i', j')$ are connected by an edge if and only if $|i - i'| + |j - j'| = 1$ (Fig. 1(a)). The edge which connects $v$ and $v'$ is directed from $v$ to $v'$ if and only if the distance from $v'$ to $r$ is smaller than that from $v$ to $r$ (Fig. 1(b)). A network defined on a grid graph is called grid network. We assume throughout this paper that, in networks we are concerned with, the capacities of all arcs have the same value $c \in \mathbb{R}_+$ and the transit times of all arcs take the same value $\tau \in \mathbb{Z}_+$. Notice that we define $c$ and $\tau$ as not a function but an integer here. From this assumption, we use the notation $\mathcal{N} = (D = (V, A), b_v, V^*)$ for simplicity by omitting the capacity function and the transit time function. In addition to the above assumption for the capacities and the transit times of arcs, we assume a sink is an inner vertex, i.e. the indegree of a sink is four (the other case can be similarly treated).

Given a grid network $\mathcal{N} = (D = (V, A), b_v, V^* = \{r\})$, we consider the quickest flow problem QF formally defined as follows:

$$\text{QF: minimize } T(\text{AT}_r^f) \text{ subject to } f \in \mathcal{F}_\mathcal{N}$$

where $\text{AT}_r^f$ is the arriving-table at the sink $r$ with respect to $f$.

For any vertices $v$ and $w$ such that there exists a directed path from $v$ to $w$, we define $l(v, w)$ as the sum of transit times of arcs on the path. Vertex set $V$ is
partitioned into layers according to the distance from \( r \). Thus, a directed graph \( D \) can be viewed as a \textit{layered graph}. A layered graph \( D = (V, A) \) is a directed graph consisting of several layers which partition \( V \) into subsets \( V^0(= \{ r \}), V^1, V^2, \ldots \) such that vertices \( v \in V^i \) and \( w \in V^j \) are connected by a directed arc \((v, w)\) only if \( i - j = 1 \), and \( V^p \) (\( p \)-th layer) denotes the set of all of vertices satisfying \( l(v, r) = pr \) (Fig. 1(c)). A network defined on a layered graph is called a \textit{layered network}.

![Fig. 1.](image)

3 Quickest Flow Problem for Grid Graphs

In this section, we consider the quickest flow problem for a grid network \( \mathcal{N} = (D = (V, A), b_v, V^* = \{ r \}) \). First we propose the overall idea of our algorithm.

Our algorithm benefits from the structure of a grid graph. Let \( CH_r = \{u_1, u_2, u_3, u_4\} \). By the way of directing arcs of a grid graph, we can decompose \( V \) into eight subsets, \( U_1, U_2, U_3, U_4 \) and \( W_1, W_2, W_3, W_4 \) as in Fig. 2 where \( U_i \) denotes the set of vertices on horizontal or vertical axis whose supplies are all sent to sink \( r \) through arc \((u_i, r)\) and \( W_i \) denotes the set of vertices whose supplies are sent to sink \( r \) through either \((u_i, r)\) or \((u_{i+1}, r)\) (we assume throughout the paper that the index \( i \) is given as \((i \mod 4) + 1\)). Here let \( H_i, i = 1, 2, 3, 4 \) be a subgraph induced by \( W_{i-1} \cup U_i \cup W_i \cup \{ r \} \). For an optimal dynamic flow \( f \) for problem QF, it can be decomposed into four flows \( f_i, i = 1, 2, 3, 4 \) such that each \( f_i \) represents the flow of supplies which reaches \( r \) through arc \((u_i, r)\). The subflow \( f_i, i = 1, 2, 3, 4 \) induces a rooted graph \( D_i \) such that its vertex set and arc set are defined as those which a positive amount of \( f_i \) passes through. Notice that \( D_i \) contains an arc \((u_i, r)\) and its vertex set is a subset of \( W_{i-1} \cup U_i \cup W_i \cup \{ r \} \) (\( D_i \) is clearly a subgraph of \( H_i \)).

The proposed algorithm is based on the following four ingredients.
Theorem 1. There exists an optimal dynamic flow $f$ such that $f_i$ and $f_j$ does not share any arc for every $i \neq j$.

Now suppose that for every $v \in W_i$ with $i = 1, 2, 3, 4$, the amount of supply (denoted by $b_{v,i}$ and $b_{v,i+1}$ respectively) which reach $r$ via arcs $(u_i, r)$ and $(u_{i+1}, r)$ respectively are fixed.

Theorem 2. There exists a subgraph $H'_i$ of $H_i$ which spans $W_{i-1} \cup U_i \cup W_i \cup \{r\}$ for $i = 1, 2, 3, 4$ such that $H'_i$ are arc disjoint for $i \neq j$.

Notice that that arc-disjoint subgraph $H'_i$ are not uniquely determined.

Theorem 3. Let us consider dynamic flow problems $QF_i$ defined on $H'_i$ such that the supply of $v \in W_{i-1} \cup U_i \cup W_i$ is $b_{v,i}$. The optimal objective value for $QF_i$ for every $i$ does not depend on the choice of arc-disjoint subgraphs $H'_i$, but remains the same.

From these facts, when for every $v \in W_i$ and every $i$ with $i = 1, 2, 3, 4$, the amount of supply at $v$ which flow through $(u_i, r)$ and $(u_{i+1}, r)$, respectively are fixed, an optimal flow of $QF$ can be found by independently obtaining an optimal flow $f^*_i$ for $QF_i$ for each $i$. Since the subgraph $H'_i$ can be chosen as a rooted tree as will be seen in Fig 3 (this fact clearly proves Theorem 2), the solution of $QF_i$ can be given by simply specifying the supply at each $v \in W_{i-1} \cup U_i \cup W_i$. Therefore, the problem $QF$ reduces to finding an optimal allocation of $b_v$ to $b_{v,i}$ and $b_{v,i+1}$ for each $v \in W_i$ with $i = 1, 2, 3, 4$, and we call this problem the optimal allocation problem for supplies. Moreover, we prove the following theorem. Consequently, we can solve the quickest flow problem for grid networks with uniform arc capacity efficiently.

Theorem 4. The optimal allocation problem for supplies can be transformed into the min-max resource allocation problem under network constraints [6–8].
problem for supplies, and (2) the second phase is to reduce the optimal allocation problem for supplies to the min-max resource allocation problem under network constraints [6–8].

3.1 Reduction the quickest flow to the optimal allocation problem for supplies

In this subsection, we prove that the quickest flow problem QF can be reduced to the optimal allocation problem for supplies. From the above discussion, the reduction is done by proving Theorem 1 and Theorem 3. Theorem 3 is proved after showing Lemma 1 and Lemma 2. We prove these lemmas by using properties of flow-tables. Thus, before proving these lemmas we introduce operators concerning flow-tables: shifting, and ceiling [2], and we will then show some basic properties of those operations.

Definition 1 (table shifting). For any flow-table FT, τ ∈ Z+ and θ ∈ Z+, we define Sτ(FT) as follows: Sτ(FT)(θ) = 0 if θ < τ and Sτ(FT)(θ) = FT(θ − τ) if θ ≥ τ.

It is easy to see that for any flow-tables FT1, FT2 and τ1, τ2 ∈ Z+, Sτ1(FT1 + FT2) = Sτ1(FT1) + Sτ2(FT2) and Sτ1+τ2(FT1) = Sτ1(Sτ2(FT1)) hold. From the above definitions and (4), (5) can be rewritten as

\[ AT_v = B_v + \sum_{e \in \Delta^-(v)} S_{\tau(e)}(ST_e). \] (6)

Definition 2 (table ceiling). For any flow-table FT and c ∈ R+, [FT]c is a flow-table obtained by carrying over the excess of FT(θ) (i.e. FT(θ) − c) to FT(θ + 1) in the order of θ = 0, 1, . . .

Here we show two facts concerning flow-tables. We use Fact 1 to prove that we need a single ceiling operation to calculate the minimum completion time of a layered network with single sink whose in-degree is one.

Fact 1 For any flow-tables FT1, FT2 and for any c ∈ R+, [[FT1]c + FT2]c = [FT1 + FT2]c.
Fact 2 For any flow-table $FT$, $c \in R^+$ and $\tau \in Z^+$, $S_\tau(FT)_c = [S_\tau(FT)]_c$ holds.

Here it should be noted that given a network $N = (D = (V,A), b_v, V^*)$ and $AT_v$ for a $v \in V$ and the distribution of $AT_v$ to each $e \in \Delta^+(v)$ as $AT^v_e$, we will assume $ST_\tau = [AT^v_e]_c$ holds for any $e \in \Delta^+(v)$ in order to attain an optimal solution [5]. Thus, throughout this paper, we restrict the set of feasible dynamic flows to those which satisfy this condition.

Let us now return to Theorem 3 again. Here we consider a layered network $L = (\tilde{D} = (\tilde{V}, \tilde{A}), \tilde{b}_v, \tilde{V}^*)$ with $|\Delta^-(\tilde{r})| = 1$, where $\tilde{V}_p = \{v \in \tilde{V} | l(v, \tilde{r}) = p\tau\}$, $\tilde{m}$ is the number of layers in $L$, $\tilde{B}_v$ denotes the extension of $b_v$ to flow-table, $\tilde{B}_p = \sum_{v \in \tilde{V}} \tilde{B}_v$ for any $p \in \{1, 2, \ldots, \tilde{m}\}$ (Fig. 5). We show the following

![Layered network L](image)

Fig. 5. Layered network $L$

lemmas concerning $L$. Lemma 1 shows the relationship between the arriving-tables of vertices whose level is $p$ and those of vertices whose level is $p + 1$.

**Lemma 1.** For any feasible flow of $L$ we have

$$[\sum_{v \in \tilde{V}_p} AT_v]_c = [\tilde{B}^p + S_\tau(\sum_{u \in \tilde{V}_{p+1}} AT_u)]_c.$$  

*Proof.* Consider $AT_v$ for $v \in \tilde{V}_p$. Then

$$AT_v = \tilde{B}_v + \sum_{u \in CH_v} S_\tau(ST_{(u,v)})$$  

(7)

holds by (6). Here we define $\{AT^u_{(u,v)} | v \in PA_u\}$ as a distribution of $AT_u$ for $u \in \tilde{V}_{p+1}$ such that $\sum_{v \in PA_u} AT^u_{(u,v)} = AT_u$ and $ST_{(u,v)} = [AT^u_{(u,v)}]_c$ hold. It is clear that such distribution always exists. Thus, we have

$$[\sum_{v \in \tilde{V}_p} AT_v]_c$$

$$= [\sum_{v \in \tilde{V}_p} (\tilde{B}_v + \sum_{u \in CH_v} S_\tau(ST_{(u,v)}))]_c \text{ (by (7))}$$

$$= [\sum_{v \in \tilde{V}_p} \tilde{B}_v + \sum_{v \in \tilde{V}_p} \sum_{u \in CH_v} S_\tau(ST_{(u,v)})]_c$$

$$= [\tilde{B}^p + \sum_{u \in \tilde{V}_{p+1}} \sum_{v \in PA_u} S_\tau(ST_{(u,v)})]_c$$

$$= [\tilde{B}^p + \sum_{u \in \tilde{V}_{p+1}} \sum_{v \in PA_u} S_\tau([AT^u_{(u,v)}]_c)]_c \text{ (by } ST_{(u,v)} = [AT^u_{(u,v)}]_c)$$


Let us decompose a \( \hat{f} \) into an arc \( e \) such that both \( f_i \) and \( f_j (i \neq j) \) share. If there is such an arc \( e \), it is called a mixed arc with respect to \( f \), and such flow \( f \) is called a mixed flow.

Proof. (Theorem 1) Let us consider an optimal dynamic flow \( \hat{f} \), and assume that it is a mixed flow. Let us decompose \( \hat{f} \) into \( f_i, i = 1, 2, 3, 4 \), and \( D_i \) for \( f_i \) be \( D_i \equiv (V, A_i) \).

From the proof assumption, \( A_i \cap A_j \neq \emptyset \) for some \( i \neq j \). Let us define a network \( \hat{N} \) for \( D_i \) such that the arc capacity and the transit time of all \( e \in \hat{A}_i \) remain the same as the original problem, and the initial supply of \( v \in V \) is equal to \( b_{v,i} \). Now, it holds for \( i = 1, 2, 3, 4 \) that \( f_i \) is a feasible dynamic flow.
of \( \tilde{N}_i \). Here let us define a network \( N_i \) such that initial supply of vertices, the arc capacity and transit time are the same as \( \tilde{N}_i \) and the underlying graph is \( H'_i \equiv (V_i, A_i) \) as the one shown in Fig. 3. Notice that \( \tilde{V}_i \subseteq V_i \) holds.

If we independently consider the dynamic flow problems for \( N_i, i = 1, 2, 3, 4 \), the optimal objective values for \( N_i \) and \( \tilde{N}_i \) are the same for each \( i = 1, 2, 3, 4 \) from Lemma 2. Let \( f^*_i \) for \( i = 1, 2, 3, 4 \) denote an optimal dynamic flow for \( N_i, i = 1, 2, 3, 4 \) respectively, and let \( f^*_i \) for \( i = 1, 2, 3, 4 \) denote an optimal dynamic flow for \( \tilde{N}_i, i = 1, 2, 3, 4 \) respectively. Then, we have

\[
\Theta(f^*_i) = \Theta(f^*_i) \leq \Theta(\tilde{f}_i).
\]

This proves the theorem because \( N_i \) with \( i = 1, 2, 3, 4 \) are arc-disjoint. \( \Box \)

Let us consider the four arc-disjoint networks \( N_i \) defined in the proof of Theorem 1. For each \( v \in W_i, i = 1, 2, 3, 4 \), initial supply \( b_v \) at \( v \) goes through either \( N_i \) or \( \tilde{N}_{i+1} \) to reach \( r \). Let \( b_{v,i} \) and \( b_{v,i+1} \) denote the amount of supplies which go to \( r \) through \( N_i \) and \( \tilde{N}_{i+1} \) respectively. We say that \( b_{v,i} \) and \( b_{v,i+1} \) are assigned to \( N_i \) and \( \tilde{N}_{i+1} \), respectively. For a vertex \( v \) on \( U_i \) with \( i = 1, 2, 3, 4 \), all the supply goes to \( v \) using \( N_i \). In this case, we say that \( b_{v,i} \) is assigned to \( N_i \). In general, let \( B_v \) and \( B_{v,i} \) denote the flow-table corresponding to \( b_v \) and \( b_{v,i} \) respectively. Then the quickest flow problem QF can be written as follows:

\[
\text{minimize} \quad \max_{i=1,2,3,4} \sum_{v \in N_i} S_i(v)(B_{v,i})_{c}
\]

subject to \( B_{v,i} = B_v \) and \( B_{v,j} = 0, j \neq i \) for \( v \in U_i \),

\[
B_{v,i} + B_{v,i+1} = B_v \quad \text{and} \quad B_{v,j} = 0, j \neq i, i+1 \quad \text{for} \quad v \in W_i,
\]

For every \( p \) and \( i \), let \( b^p_i = \sum_{v \in W_i \cap (W_{i+1} \cup W_{i+2})} b_{v,i} \) which represents the amount of supply of vertices at level \( p \) assigned to \( N_i \), and let \( B^p_i \) denote its flow-table extension. Then from Lemma 2, the completion time for \( N_i \) is expressed as \( T((\sum_{i=1}^m S_{pr}(B^p_i))_c) \). Therefore, the assignment of \( b_v \) of a particular vertex \( v \in V^p \cap (W_i \cup W_{i-1}) \) does not affect the minimum completion time for \( N_i \) but only the total amount \( b^p_i \) assigned to \( N_i \) does affect it. From this observation, we contract the set \( V^p \cap W_i \) into a single vertex \( w^p_i \) for each \( p \) with \( 1 \leq p \leq m \) and \( i \) with \( 1 \leq i \leq 4 \) such that the initial supply of \( w^p_i \) is equal to \( \sum_{v \in V^p \cap W_i} b_v \) which is simply denoted by \( a^p_i \). Let \( a^p_i \) denote a single vertex corresponding to \( V^p \cap U_i \). The initial supply of \( a^p_i \) is denoted by \( g^p_i \). Let the assignment of \( a^p_i \) to \( N_i \) and \( \tilde{N}_{i+1} \) be \( a^{p,i} \) and \( a^{p,i+1} \), respectively. Therefore, defining the flow-table \( FT_i \) such as

\[
FT_i(pr) = g^p_i + a^{p,i} + a^{p-1,i}, \quad p = 1, 2, \ldots, m,
\]

the minimum completion time of \( N_i \) is equal to \( T([FT_i])_c \). Here we introduce the following theorem to calculate \( T([FT_i])_c \) efficiently. Lemma 3 shows that we can calculate \( T([FT_i])_c \) without operating table-ceiling explicitly.

**Lemma 3.** For any flow-table \( FT \) and \( c \in R_+ \),

\[
T([FT])_c = \max_{0 \leq \theta \leq T(FT)} \left[ \frac{c \theta + \sum_{t=\theta}^{T(FT)} FT(t)}{c} \right] - 1.
\]
Proof. We give a sketch of the proof. The idea is to prove the time that satisfies
\[
\max_{0 \leq \theta \leq T(FT)} \{ c\theta + \sum_{t=0}^{T(FT)} FT(t) \} \text{ is equal to }
\]
\[
\max \left\{ \theta \in \mathbb{Z}_+ \mid \sum_{t=0}^{\theta} FT(t) = \sum_{t=0}^{\theta} [FT]_c, \theta < T([FT]_c) \right\} + 1.
\]
This claim can be proved by the properties of any flow-table \(FT\) as follows:
\[
\sum_{t=0}^{\theta} FT(t) \geq \sum_{t=0}^{\theta} [FT]_c(t) \text{ for any } \theta \in \mathbb{Z}_+, \quad \text{and}
\]
\[
\sum_{t=0}^{\theta} FT(t) = \sum_{t=0}^{\theta} [FT]_c(t) \text{ for any } \theta \in \mathbb{Z}_+ \text{ with } [FT]_c(\theta) < c.
\]
Thus, from Lemma 3 and (9), \(QF\) can be reduced to the following problem \(QF'\).

\[
\begin{align*}
QF' \quad & \minimize \max_{1 \leq i \leq 4} \max_{1 \leq p \leq m} \{ c p \tau + \sum_{k=p}^{m} FT_i(k \tau) \}
\end{align*}
\]

3.2 Reduction to min-max resource allocation problem under network constraints

The problem \(QF'\) can be reduced to the min-max resource allocation problem under network constraints as will be shown below. This problem is a kind of min-max flow problem with multiple sources and sinks in a static network [6–8] which is defined as follows. Suppose we are given a network with multiple sources and sinks such that a fixed amount of supply is associated with each source, and the cost function \(\gamma_t(x_t)\) which is nondecreasing in \(x_t\) is associated with each sink \(t\) where \(x_t\) denotes the amount of flow entering \(t\). Then the problem asks to find a (static) flow that minimizes the maximum of the cost functions of sinks.

We will explain how we construct a (static) network (see Fig. 6) for which finding an optimal solution for the min-max resource allocation problem produces an optimal solution of problem \(QF'\). The network to be constructed consists of \(m\) components \(C^1, C^2, \ldots, C^m\). Each component \(C^p\) except \(C^m\) has four
layers while $C^m$ has three layers. The first layer of each component $C^p$ has eight sources which correspond to vertices $w_{1,i}^p, u_{1,i}^p, i = 1, 2, 3, 4$ defined in the previous subsection. The second and third layers consists of four vertices denoted by $v_{1,i}^p, v_{2,i}^p, i = 1, 2, 3, 4$. The fourth layer consists of a single vertex $v_{3}^p$. The connection between the layers are as shown in Fig. 7(a). Vertex $u_{1}^p$ is connected to $v_{1,1}^p$, and $v_{1,1}^p$ for each $i = 1, 2, 3, 4$. The actual cost function associated with the min-max resource allocation problem here is defined in the previous subsection. The cost function associated with each sink $v_{m,i}^p, i = 1, 2, 3, 4$ is the amount of the excess carried.

![Figure 7](image-url)

**Fig. 7.** (a)p-th component $C^p$ (b)m-th component $C^m$ (c)p-th bridges

The meaning of the capacity $c_T$ on the arcs from the second to the third layer in $C^p$ with $1 \leq p \leq m - 1$ is as follows. Let us consider $FT_i$ of (9) and perform ceiling operation to obtain the completion time of $N_i$. Let us recall that in performing the ceiling operation the amount of supply carried over to time $p'\tau$ plus the amount $FT_i(p'\tau)$ will then be carried over to $FT_i((p'\tau + 1), FT_i(p'\tau + 2), \ldots$. If the amount of supply carried over to time $p'\tau$ plus the amount $FT_i(p'\tau)$ is less than or equal to $c_T$, the amount $FT_i((p'\tau + 1)\tau)$ is attained for $p > p'$. Thus, a positive flow going through a bridge stands for the situation that the amount of supply carried over to time $p'\tau$ plus the amount $FT_i(p'\tau)$ is larger than $c_T$ and thus a positive amount of flow will be carried over to $FT_i((p' + 1)\tau)$. The cost function associated with the min-max resource allocation problem here associated with each sink $v_{m,i}^p, i = 1, 2, 3, 4$ is the amount of the excess carried.
over to time \( m\tau \) when performing ceiling operation to \( FT_i \) which is equivalent to \( \max_{1 \leq p \leq m} \{ c\tau p + \sum_{k=p}^{m} FT_i(k\tau) \} = cm\tau \). Since \( cm\tau \) is constant, the min-max resource allocation problem defined in this subsection solves problem \( QF' \).

It is known that the min-max resource allocation problem for the network with \(|V|\) vertices, \(|A|\) arcs and \(|T|\) sinks can be solved in \( O(|T|(|V||A| \log |V| + |T| \log \frac{M}{|T|})) \) time where \( M \) denotes the sum of supplies \([6–8]\). The second term in the parenthesis, i.e., \( O(|T| \log \frac{M}{|T|}) \), is the time required to solve the resource allocation problem without the network constraints. Since our cost function associated with \( v_{2i}^m, i = 1, 2, 3, 4 \) is linear, we can reduce the time to \( O(|T|) \) (the details are omitted). In our case, \(|T|\) is constant and \(|V| = O(\sqrt{n})\), \(|A| = O(\sqrt{n})\), thus the running time becomes \( O(n \log n) \).

4 Conclusion

We have presented an \( O(n \log n) \) time algorithm for the quickest flow problem in a grid network with uniform arc capacity. The algorithm proposed in this paper can be extended to a general layered network \( \mathcal{N} \) such that (1) the transit time from a vertex \( v \) to a sink \( r \) dose not depend on the choice of a path, and (2) the underlying layered graph \( D \) can be decomposed into arc-disjoint layered graphs \( D_1, D_2, \ldots, D_k \) which spans \( V_1, V_2, \ldots, V_k \) respectively, where \( CH_r = \{v_1, v_2, \ldots, v_k\} \) and \( V_i \) is the set of vertices from which \( v_i \) is reachable in \( D \).

Thus, the result can also be generalized to the case where the arc capacity is a multiple of \( c \) by regarding the arc as multiple ones as long as the resulting layered graph satisfies the requirement just mentioned above.

References