

Article

A Pathfinding Problem for Fork-Join Directed Acyclic Graphs with Unknown Edge Length

Kunihiko Hiraishi 

School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai,
Nomi 923-1292, Ishikawa, Japan; hira@jaist.ac.jp; Tel.: +81-761-51-1281

Abstract: In a previous paper by the author, a pathfinding problem for directed trees is studied under the following situation: each edge has a nonnegative integer length, but the length is unknown in advance and should be found by a procedure whose computational cost becomes exponentially larger as the length increases. In this paper, the same problem is studied for a more general class of graphs called fork-join directed acyclic graphs. The problem for the new class of graphs contains the previous one. In addition, the optimality criterion used in this paper is stronger than that in the previous paper and is more appropriate for real applications.

Keywords: shortest path problem; planning problem; online algorithm



Citation: Hiraishi, K. A Pathfinding Problem for Fork-Join Directed Acyclic Graphs with Unknown Edge Length. *Algorithms* **2021**, *14*, 367. <https://doi.org/10.3390/a14120367>

Academic Editor: Frank Werner

Received: 18 December 2021

Accepted: 15 Decembre 2021

Published: 17 Decembre 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In a previous paper by the author [1], a pathfinding problem for directed trees is studied under the following situation: each edge has a nonnegative integer length, but the length is unknown in advance and should be found by a procedure whose computational cost becomes exponentially larger as the length increases. Such a situation arises in an operation synthesis problem for reconfigurable cloud computing systems [2,3]. This problem is described as follows. A typical reconfigurable cloud computing system consists of multiple physical servers interconnected via network switches. Servers may have different computing resources such as CPU, memory, and hard disk drives. Multiple virtual machines are running under the virtual machine monitor in each server, and application software runs on each virtual machine. Such an arrangement of virtual machines, operating systems, and application software on physical servers is called a configuration. Then, the problem is to find a sequence of operations that leads the system from the initial configuration to a given goal configuration. Since the problem becomes harder as the length of the operation sequence becomes longer, implementing subgoals is proposed to shorten the total computation time. The number of necessary operations between two subgoals is not known in advance and can be known by applying some search procedure. This situation is formulated as the above graph problem.

In this paper, the same problem is studied for a more general class of graphs called fork-join directed acyclic graphs. We call this problem PFJUEL (Pathfinding in an FJ-DAG with Unknown Edge Length). There is a remaining problem in the previous paper, which is the optimality criterion used for evaluating the solution method. The optimality criterion used in this paper is stronger than that in the previous paper and is more appropriate for real applications. Moreover, the argument to derive the optimality becomes clearer for this generalized class of graphs.

Solution methods to the problem involve a procedure whereby the next action to be taken is determined by the current knowledge of the target. We call this type of procedure a strategy. Such a problem has been studied as the planning problem in artificial intelligence, and there are a variety of algorithms in this area. Typical algorithms are the A^* algorithm [4] and its variants such as [5–7]. In the research field of graph algorithms, many algorithms have been proposed for solving shortest path problem in varieties of situations. Well-known

algorithms for finding a shortest path, such as Dijkstra's algorithm and the Bellman–Ford algorithm, cannot be applied because the length of each edge is unknown in advance. There exist studies on graphs with uncertainty. Algorithms to solve such problems with uncertainty include online algorithms [8]. The Canadian traveler problem [9,10] is one of these algorithms. In the Canadian traveler problem, whether each edge (v_i, v_j) is available or not is known only when the vertex v_i is visited. Solution methods to this problem are strategies, and the cost of a strategy is defined as the sum of the lengths of all edges traversed. Online shortest path problems for graphs with uncertainty are also studied [11], and there are many applications such as route finding in transit networks [12]. In these problems, the weight of each edge can change in an arbitrary way. Such a situation occurs in real transit networks because of traffic jams. There are several differences between PFJUEL and the existing online graph problems with uncertainty. Although most of the graph problems with uncertainty are defined in stochastic domain [8,13–15], PFJUEL is defined in a fully deterministic way. Moreover, the edge length does not change (but is not known in advance). In some online shortest path problems, the existence of an agent that traverses the network is assumed. The agent makes decisions based on the information around the agent and its history. In PFJUL, no agent is assumed, and the operation can be done for any point in the graph. The evaluation method of the algorithm is also different. To evaluate the performance of online algorithms, the competitive ratio [16] is often used. This is the ratio between the performance of an online algorithm and that of an offline algorithm. From the motivated examples shown in [2,3], we assume that the computational cost of the procedure for finding the length of each edge is dominant in the total computational cost. Therefore, it suffices to consider the number of procedure calls for the evaluation. The detail of the evaluation method will be described later.

The paper is organized as follows. In Section 2, the problem studied in the paper is formally described. In Section 3, how to evaluate solution methods is presented. The current knowledge on the graph is summarized as the estimate, and the optimality of a method is evaluated by the estimate the method finally gives. In Section 4, we define a special form of estimates that gives the optimal estimate. In Section 5, a solution method to the problem is presented. The method gives an optimal solution under an assumption. For the case without the assumption, the difference between the obtained solution and the optimal one is evaluated. Section 6 presents the conclusion.

2. Problem Formulation

We first formally define a class of graphs called fork-join directed acyclic graphs (FJ-DAGs). An FJ-DAG is a directed acyclic graph $G = (V, E)$, where V is the set of vertices and $E \subseteq V \times V$ is the set of edges, with two special vertices the top vertex $v^t \in V$ and the bottom vertex $v^b \in V$. We denote $G = (V, E, v^t, v^b)$ to indicate the two special vertices.

Definition 1. FJ-DAGs are recursively defined as follows:

1. A single vertex $G = (\{v\}, \emptyset, v, v)$ is an FJ-DAG;
2. Let $G_i = (V_i, E_i, v_i^t, v_i^b)$, $i = 1, \dots, m$ be FJ-DAGs. Then, the graph

$$G = \left(\bigcup_i^m V_i \cup \{v^t, v^b\}, \bigcup_i^m E_i \cup \{(v^t, v_i^t), (v_i^b, v^b) \mid i = 1, \dots, m\}, v^t, v^b \right)$$

is an FJ-DAG, where v^t and v^b do not belong to any V_i . Each G_i is called a child of G ;

3. No other graphs defined as above are not FJ-DAGs.

Each edge $e_i^t = (v^t, v_i^t)$ is called a top edge of G and each edge $e_i^b = (v_i^b, v^b)$ is called a bottom edge of G . An FJ-DAG has a nested structure. We introduce the level of an FJ-DAG G , denoted by $level(G)$, as follows: (i) the level of a single vertex G is 0; (ii) if $G_i, i = 1, \dots, m$ are children of G , then $level(G) = \max_i level(G_i)$. Any FJ-DAG contained in an FJ-DAG G as a subgraph is called a sub FJ-DAG of G . Figure 1 shows an FJ-DAG,

where the level of each sub FJ-DAG is indicated. We also define the level of each edge. The level of an edge is defined as the level of a sub FJ-DAG that contains the edge as a top or bottom edge. Such a class of graphs is well studied in the research field of queuing networks since it appears in various applications such as parallel computing and flexible manufacturing systems [17].

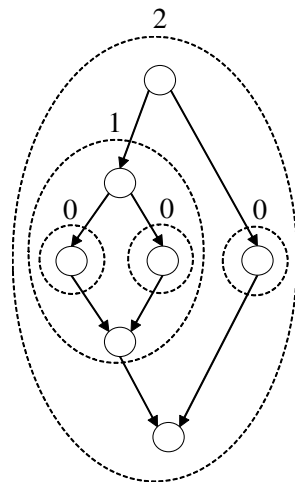


Figure 1. An FJ-DAG. The level of each sub FJ-DAG is indicated.

In an FJ-DAG $G = (V, E, v^t, v^b)$, a path from v^t to v^b is called a goal path of G . Now, we define a graph problem, Pathfinding in an FJ-DAG with Unknown Edge Length (PFJUEL), as follows:

Given

- An FJ-DAG $G = (V, E, v^t, v^b)$;
- Oracle $search_c(v_i, v_j)$: given two vertices v_i, v_j and a nonnegative integer c , the oracle answers whether the length of edge (v_i, v_j) is less than or equal to c or not.

Assumption

- Each edge has a nonnegative integer length, but the length is unknown in advance and should be found by calling the oracle.

Find

- A shortest goal path.

PFJUEL is an online problem since the length of each edge is known only by calling oracles. As we have mentioned in the introduction, any procedures that solve such an online problem should be adaptive; i.e., the next action (i.e., an oracle call) to be taken is determined by the current knowledge of the target. Such a procedure is called a strategy. Roughly speaking, the objective of PFJUEL is to find a strategy that is optimal for the total cost of oracle calls. The formal statement of an optimal strategy to PFJUEL is given after we introduce the required concepts and terminology.

We note that PFJUEL is a generalization of the problem PSTUEL studied in [1]. The targets of PSTUEL are directed trees. Given a directed tree, we can make an FJ-DAG such that the shortest goal path of the FJ-DAG contains the shortest goal path of the directed tree, where a goal path of a tree is a path from the root to a leaf. To do this, we add edges with length 0 so that the tree becomes an FJ-DAG (Figure 2).

At some step of a strategy for PFJUEL, the set of edges E is partitioned into two sets E_K and E_U , where the length $w(v_i, v_j)$ of each edge $(v_i, v_j) \in E_K$ is already known, and the length of each edge in E_U is unknown. An edge (v_i, v_j) moves from E_U to E_K when its length has been found. Moreover, for each edge $(v_i, v_j) \in E_U$, a lower bound $\hat{w}(v_i, v_j)$ of the edge length is obtained by previous oracle calls. The initial lower bound is 0, and if $search_c(v_i, v_j)$ returns “no”, then $c + 1$ is set to $\hat{w}(v_i, v_j)$.

Definition 2. A strategy that does not call $search_c(v_i, v_j)$ for $c > w(v_i, v_j)$ is called conservative.

In conservative strategies, an edge (v_i, v_j) moves from E_U to E_K when $search_c(v_i, v_j)$ returns “yes” for $c = w(v_i, v_j)$. Since calling the oracle for a large c is dominant in the total computational cost, we concentrate on conservative strategies only. In this class of strategies, calling $search_c(v_i, v_j)$ for all $c = 0, 1, \dots, w(v_i, v_j)$ in this order is mandatory to know the correct edge length.

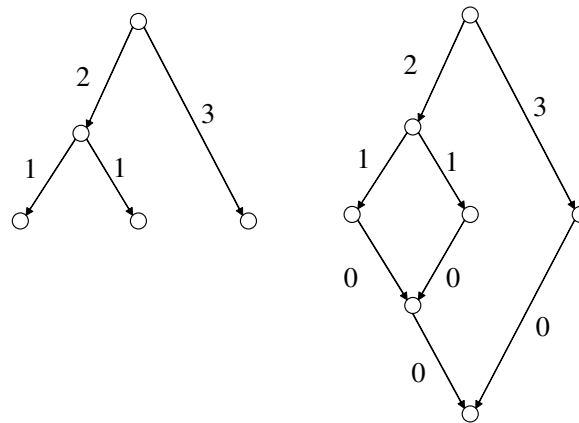


Figure 2. A tree and the constructed FJ-DAG.

3. Estimate and Characteristic Vector

For conservative strategies, the current knowledge about the graph is represented by a pair $F = (F_K, F_U)$, where F_K is the set of all pairs $\langle (v_i, v_j), w(v_i, v_j) \rangle$ with $(v_i, v_j) \in E_K$, and F_U is the set of all pairs $\langle (v_i, v_j), \hat{w}(v_i, v_j) \rangle$ with $(v_i, v_j) \in E_U$. We call such F an estimate of G . Let \mathcal{F} denote the set of all estimates. A strategy for PFJUEL is formally defined as a mapping \mathcal{S} from \mathcal{F} to $E_U \times \mathbb{N}$ (this definition of strategies means that strategies are deterministic; i.e., any strategy gives the same response to the same estimate). Based on the previous results on oracle calls, strategy \mathcal{S} gives the next edge in E_U and nonnegative integer c for which the oracle is called. The current lower bound of the path length is given by the sum of $\hat{w}(v_i, v_j)$ for all edges (v_i, v_j) on the path. We call this the estimated length of the path.

When a shortest goal path is found, the estimate has to satisfy the following requirements in order to guarantee the correctness of the result.

Definition 3. An estimate is called terminal if the following two conditions hold:

- R1. All the edges on a shortest goal path π^* are in E_K ;
- R2. For any goal path other than π^* , its estimated length is no less than the length of π^* .

In what follows, oracle calls that return “no” are called fail calls and oracle calls that return “yes” are called success calls. Given an estimate F , we define a vector $\#_F = [a_0, a_1, \dots, a_s]$ called the characteristic vector of F , where each a_k is the number of edges with $\hat{w}_F(v_i, v_j) = k + 1$, and s is the largest integer such that $a_s \neq 0$. In conservative strategies, the number of fail calls $search_c(\cdot, \cdot)$ is given by $\sum_{i=c}^s a_i$, because if $search_c(v_i, v_j)$ fails, then $search_{c'}(v_i, v_j)$ fails for all $0 \leq c' < c$.

To define the optimality of strategies, we introduce a lexicographical order \preceq on the set of characteristic vectors. For two characteristic vectors $\#_{F_1} = [a_0, a_1, \dots, a_s]$ and $\#_{F_2} = [b_0, b_1, \dots, b_t]$, we write $\#_{F_1} \prec \#_{F_2}$ if (i) $s < t$ or (ii) $s = t$ and $a_j < b_j$ holds for $j = \max\{i \in \{0, 1, \dots, s\} \mid a_i \neq b_i\}$, and define $\#_{F_1} \preceq \#_{F_2}$ to be $\#_{F_1} \prec \#_{F_2}$ or $\#_{F_1} = \#_{F_2}$.

Definition 4. A terminal estimate F is called optimal if for any other terminal estimate F' , $\#_F \preceq \#_{F'}$ holds.

The characteristic vectors do not reflect the number of success calls, since the lower bound is not updated for any success call. In the conservative strategies, however, every success call $search_c(v_i, v_j)$ for $c = w(v_i, v_j)$ is preceded by a fail call $search_{c-1}(v_i, v_j)$. This means that the number of success calls $search_c(\cdot, \cdot)$ is no greater than the number of fail calls $search_{c-1}(\cdot, \cdot)$. For this reason, the optimality defined above can be a measure of the total computational cost.

We have assumed that the cost of calling oracles is dominant in the total computation time and it becomes exponentially larger as the length c increases. Since we give no exact computational cost of oracle calls, the optimality of a strategy should be defined by the number of oracle calls. Under the assumption of the computational cost of oracle calls, the above definition is reasonable. In our previous paper, we initially intended to use the same evaluation measure as that used in this paper. However, we could not prove the optimality for it and could prove the optimality for a weaker measure in which only the number of oracle calls for the largest c is considered.

If all goal paths have the same estimated length h , then the estimate is called homogeneous with length h .

Lemma 1. *Any optimal estimate is homogeneous.*

Proof. Assume that there exists an optimal estimate having at least one goal path with length $h > h^*$, where h^* is the length of the shortest goal path. Choose a goal path π with length h . π has the form

$$e_{(1)}^t \cdots e_{(k)}^t e_{(k)}^b \cdots e_{(1)}^b$$

where (i) denotes the index of a sub FJ-DAG $G_{(i)}$. If π does not share edges with any goal path with length h^* , then we choose one edge with a nonzero estimated length (since $h > h^*$, such an edge exists) and decrement the estimated length by one. Suppose that π shares an edge $e_{(i)}^t$ or $e_{(i)}^b$ with goal paths having length h^* . From the structure of FJ-DAGs, edges $e_{(1)}^t, \dots, e_{(i)}^t, e_{(i)}^b, \dots, e_{(1)}^b$ are also shared. Since π is longer than such paths with length h^* , there exists an edge $e_{(j)}^t$ or $e_{(j)}^b, j > i$ that is not shared by such goal paths and has a nonzero estimated length. Then, we decrement the estimated length of the edge by one. As a result, the length of π can be decreased by one. The estimate is still terminal after this procedure. This contradicts to the assumption that the estimate is optimal. \square

4. Canonical Estimate

In this section, we introduce a special class of estimates, called canonical estimates, that gives optimal estimates. We first introduce some terminology. An edge (v_i, v_j) is called saturated if $\hat{w}(v_i, v_j) = w(v_i, v_j)$, and is called unsaturated if $\hat{w}(v_i, v_j) < w(v_i, v_j)$. A cut is a subset of edges such that it contains exactly one edge on every goal path. Let F be the characteristic vector of an estimate. Then, the characteristic vector of a cut is the restriction of F to the edges in the cut. A cut is called maximum if its characteristic vector is maximum with regard to \preceq in all cuts. A cut is called unsaturated if it consists of unsaturated edges only. An unsaturated minimum cut is an unsaturated cut such that its characteristic vector is minimum with regard to \preceq in all unsaturated cuts. Figure 3 shows a maximum cut with characteristic vector $[1, 1, 1]$ and an unsaturated minimum cut with characteristic vector $[2, 1]$.

Definition 5. (Canonical Estimate)

Let F be an estimate of an FJ-DAG G . If $level(G) = 0$, then F is canonical. Suppose that G has children $G_i (i = 1, \dots, m)$ as shown in Figure 4. Then, F is called canonical if it is homogeneous and the following conditions hold for all $i = 1, \dots, m$:

- C1. F is canonical for G_i ;
- C2. If $e_i^t (e_i^b)$ is unsaturated, then $\hat{w}(e_i^b) \leq \hat{w}(e_i^t) + 1$ ($\hat{w}(e_i^t) \leq \hat{w}(e_i^b) + 1$) holds;

- C3. If $\text{level}(G_i) \geq 1$ and e_i^t (e_i^b) is unsaturated, then $\hat{w}(e_i^t) \geq \hat{w}(G_i)^\uparrow$ ($\hat{w}(e_i^b) \geq \hat{w}(G_i)^\uparrow$) holds, where $\hat{w}(G_i)^\uparrow$ is the largest estimated length in G_i ;
- C4. If $\text{level}(G_i) \geq 1$ and G_i has at least one unsaturated cut, then $\hat{w}(e_i^t) \leq \hat{w}(G_i)^\downarrow + 1$ and $\hat{w}(e_i^b) \leq \hat{w}(G_i)^\downarrow + 1$ hold, where $\hat{w}(G_i)^\downarrow$ is the largest estimated length in all unsaturated minimum cuts of G_i .

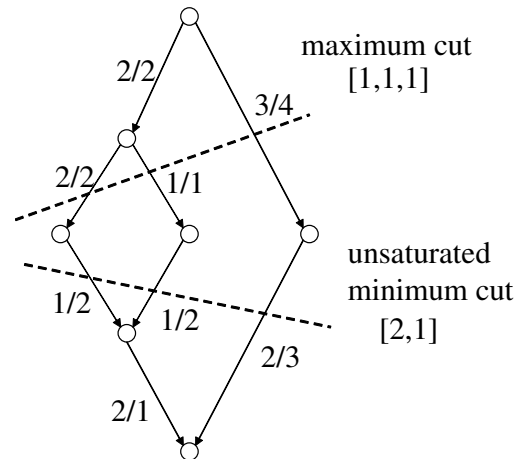


Figure 3. Maximum/minimum cut (for each edge (v_i, v_j) , $\hat{w}(v_i, v_j)/w(v_i, v_j)$ is indicated).

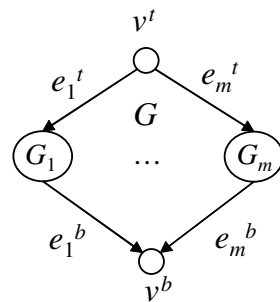


Figure 4. Structure of FJ-DAG.

Since any canonical estimate is homogeneous, all goal paths have the same estimated length. We take a canonical estimate with length h to indicate that the estimated length is h .

By reassigning the estimated lengths of some edges, the characteristic vector of a non-canonical estimate may decrease without changing its length. We show examples. The estimate shown in Figure 5a does not satisfy C2. By the reassignment shown in Figure 5b, the characteristic vector decreases. The estimate shown in Figure 6a does not satisfy C3 because the maximum estimated length of the sub FJ-DAG is 6, which is greater than the value of 3 of the bottom edge. As shown in Figure 6b, the characteristic vector decreases by adding one to the bottom edge and removing one from every edge on the maximum cut. The estimate shown in Figure 7a does not satisfy C4 because the maximum estimated length on the unsaturated minimum cut of the sub FJ-DAG is 4, and $4 + 1 = 5$ is less than the value of 6 of the bottom edge. As shown in Figure 7b, the characteristic vector decreases by removing one from the bottom edge and adding one to every edge on the unsaturated minimum cut. These operations are used to prove the optimality of canonical estimates.

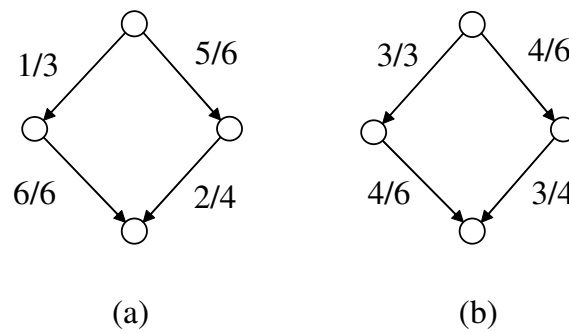


Figure 5. Reassignment for violation of C2: (a) before the reassignment, (b) after the reassignment (for each edge (v_i, v_j) , $\hat{w}(v_i, v_j)/w(v_i, v_j)$ is indicated).

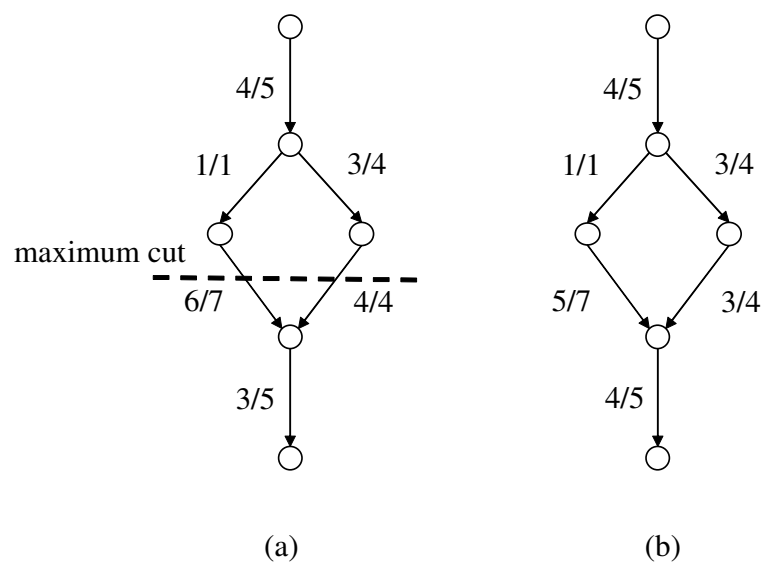


Figure 6. Reassignment for violation of C3: (a) before the reassignment, (b) after the reassignment (for each edge (v_i, v_j) , $\hat{w}(v_i, v_j)/w(v_i, v_j)$ is indicated).

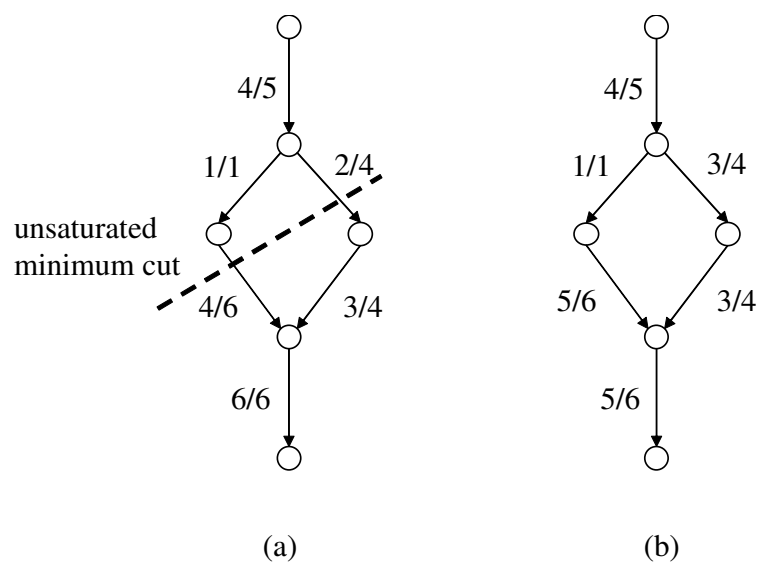


Figure 7. Reassignment for violation of C4: (a) before the reassignment, (b) after the reassignment (for each edge (v_i, v_j) , $\hat{w}(v_i, v_j)/w(v_i, v_j)$ is indicated).

Lemma 2. Let G be an FJ-DAG shown in Figure 4, and let F be a canonical estimate for G with length h . Then, for each $i = 1, \dots, m$, $\hat{w}(e_i^t)$ and $\hat{w}(e_i^b)$ are unique up to their exchange.

Proof. We first consider the case that $\text{level}(G_i) = 0$. Then, $\hat{w}(e_i^t) + \hat{w}(e_i^b) = h$ holds. If both e_i^t and e_i^b are saturated, then the value assignment is unique. If both e_i^t and e_i^b are unsaturated, then by C2, $|\hat{w}(e_i^t) - \hat{w}(e_i^b)| \leq 1$ holds and the value assignment to these edges are unique up to exchange of them. Suppose that e_i^t is saturated and e_i^b is unsaturated. Then, by C2, $\hat{w}(e_i^t) \leq \hat{w}(e_i^b) + 1$ holds. If $\hat{w}(e_i^t) < \hat{w}(e_i^b) + 1$, then this is the unique assignment to these two edges. If $\hat{w}(e_i^t) = \hat{w}(e_i^b) + 1$, then we can obtain another assignment by decrementing $\hat{w}(e_i^t)$ by one and incrementing $\hat{w}(e_i^b)$ by one. No other assignments are possible. Therefore, the value assignment is unique up to their exchange. This also holds for the case that e_i^t is unsaturated and e_i^b is saturated.

Next, we consider the case that $\text{level}(G_i) > 0$. Suppose that G_i has no unsaturated cut. Then, G_i has a goal path with saturated edges only. Let h_i be the length of such a saturated goal path. Note that h_i is uniquely determined, and all goal paths of G_i have the same estimated length since F is homogeneous. Therefore, $\hat{w}(e_i^t) + \hat{w}(e_i^b) = h - h_i$ is also unique. By the same argument as in the case $\text{level}(G_i) = 0$, we can prove the lemma.

Suppose that G_i has at least one unsaturated cut. Let $e_{(s)}^x$ (x is t or b) be an unsaturated edge in G_i such that it is on a minimum unsaturated cut C of G_i and $\hat{w}(e_{(s)}^x) = \hat{w}(G_i)^\downarrow$. Let $G_{(s)}$ be the sub FJ-DAG having $e_{(s)}^x$ as the top or bottom edge. By C3, every edge in $G_{(s)}$ is with an estimated length no greater than $\hat{w}(e_{(s)}^x)$. We claim that $G_{(s)}$ has a goal path such that all unsaturated edges on the path are with estimated length $\hat{w}(e_{(s)}^x)$. Assume that $G_{(s)}$ has no such goal paths. Then, $G_{(s)}$ has an unsaturated cut C' with a characteristic vector smaller than that of $\{e_{(s)}^x\}$. By replacing $e_{(s)}^x$ with C' in the cut C , we obtain an unsaturated cut of G_i with a smaller characteristic vector than C , which is a contradiction. Since $G_{(s)}$ is homogeneous, any goal path in $G_{(s)}$ has the same estimated length, and it is uniquely determined.

Let π be a goal path that goes through $e_{(s)}^x$. π has the form

$$e_{(1)}^t \cdots e_{(s)}^t \cdots e_{(r)}^t e_{(r)}^b \cdots e_{(s)}^b \cdots e_{(1)}^b$$

where $e_{(1)}^t = e_i^t$ and $e_{(1)}^b = e_i^b$. Note that subpaths $e_{(1)}^t \cdots e_{(s)}^t$ and $e_{(s)}^b \cdots e_{(1)}^b$ are uniquely identified. We consider the case that x is t . The case that x is b is similarly proved. We make the following observations:

- The estimated length of the subpath

$$e_{(s+1)}^t \cdots e_{(r)}^t e_{(r)}^b \cdots e_{(s+1)}^b$$

in $G_{(s)}$ is uniquely determined; [by the fact that F gives the unique estimated length to goal paths of $G_{(s)}$ as shown above]

- $\hat{w}(G_i)^\downarrow \leq \hat{w}(e_{(s)}^b) \leq \hat{w}(G_i)^\downarrow + 1$ if $w(e_{(s)}^b) > \hat{w}(G_i)^\downarrow$; $\hat{w}(e_{(s)}^b) = w(e_{(s)}^b) \leq \hat{w}(G_i)^\downarrow$ otherwise; [by the fact that $e_{(s)}^t$ is on a minimum unsaturated cut of G_i (This exclude that case $\hat{w}(e_{(s)}^t) > \hat{w}(e_{(s)}^b)$), $\hat{w}(e_{(s)}^t) = \hat{w}(G_i)^\downarrow$ as we have assumed, and C2]
- $\hat{w}(G_i)^\downarrow \leq \hat{w}(e_{(j)}^x) \leq \hat{w}(G_i)^\downarrow + 1$ if $w(e_{(j)}^x) > \hat{w}(G_i)^\downarrow$; $\hat{w}(e_{(j)}^x) = w(e_{(j)}^x) \leq \hat{w}(G_i)^\downarrow$ otherwise (x is t or b , $1 \leq j < s$); [by C3 and C4]
- $\max(\hat{w}(e_{(j)}^t), \hat{w}(e_{(j)}^b)) \leq \hat{w}(e_{(k)}^x)$ if $e_{(k)}^x$ is unsaturated (x is t or b , $1 \leq k < j \leq s$). [by C3]

The estimate for $e_{(1)}^t \cdots e_{(s)}^t$ and $e_{(s)}^b \cdots e_{(1)}^b$ that satisfies the above properties is obtained by the following procedure.

1. Assign $\hat{w}(G_i)^\downarrow$ to $e_{(j)}^x$ if $w(e_{(j)}^x) \geq \hat{w}(G_i)^\downarrow$; otherwise, make $e_{(j)}^x$ saturated (x is t or b , $j = 1, \dots, s$);

2. At this moment, the current estimate is the minimum estimate that satisfies all the properties. If the length of π is h , then halt;
3. From outmost unsaturated edges $e_{(j)}^t$ and $e_{(j)}^b$, increment the estimated length by one until the length of π reaches h .

Figure 8 shows an estimate obtained by this procedure, where $s = 5$. When the procedure terminates, $\hat{w}(e_{(1)}^t)$ and $\hat{w}(e_{(1)}^b)$ are unique up to their exchange. \square

| | | | | | | | | | | | | |
|------------------|-------------|-------------|-------------|-------------|-----------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | | G_i | | | | | | | | | | |
| length | 3 | 4 | 1 | 3 | 3 | 1 | 3 | 1 | 4 | 5 | 2 | 4 |
| | $e_{(1)}^t$ | $e_{(2)}^t$ | $e_{(3)}^t$ | $e_{(4)}^t$ | $e_{(5)}^t$ | $e_{(6)}^t$ | $e_{(6)}^b$ | $e_{(5)}^b$ | $e_{(4)}^b$ | $e_{(3)}^b$ | $e_{(2)}^b$ | $e_{(1)}^b$ |
| estimated length | 3 | 3 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 3 |
| | | | | | $\hat{w}(G_i)^{\downarrow}$ | $G_{(5)}$ | | | | | | |

Figure 8. A canonical estimate obtained by the procedure.

Proposition 1. Let G be an FJ-DAG. Any canonical estimate for G with length h has the same characteristic vector.

Proof. Let F be a canonical estimate for G , as shown in Figure 4. The proposition is obvious when $level(G) = 0$. Suppose that $level(G) > 0$. We can consider each i separately. By Lemma 2, $\hat{w}(e_i^t)$ and $\hat{w}(e_i^b)$ are unique up to their exchange. Therefore, the length of F in G_i is also the same in any canonical estimate. By the induction hypothesis, any canonical estimate for G_i has the same characteristic vector, and therefore any canonical estimate for G has the same characteristic vector as well. \square

Lemma 3. Let F be any homogeneous estimate for G with length h . Then, there exists a canonical estimate F^* with the same length h and $\#_{F^*} \preceq \#_F$.

Proof. If a homogeneous estimate is not canonical, then at least one of C2, C3, and C4 is not true in some sub FJ-DAG of G . We introduce procedures to make the estimate canonical.

Violation of C2: Let G_i be a sub FJ-DAG in which C2 is false. This means that e_i^t (e_i^b) is unsaturated and $\hat{w}(e_i^b) > \hat{w}(e_i^t) + 1$ ($\hat{w}(e_i^t) > \hat{w}(e_i^b) + 1$). Then, increment $\hat{w}(e_i^t)$ ($\hat{w}(e_i^b)$) by one and decrement $\hat{w}(e_i^b)$ ($\hat{w}(e_i^t)$) by one (see Figure 5).

The following holds:

- The estimate is still homogeneous with the same length after the update;
- By repeating this procedure, C2 eventually becomes true in G_i ;
- The characteristic vector decreases after the update;
- If C2 is true in another sub FJ-DAG $G_{(j)}$ of G , then C2 is still true in $G_{(j)}$ after the update.

Violation of C3: We assume C2 is true in any sub FJ-DAG. Let G_i be a sub FJ-DAG in which C3 is false. Then, e_i^t (e_i^b) is unsaturated and $\hat{w}(e_i^t) < \hat{w}(G_i)^{\uparrow}$ ($\hat{w}(e_i^b) < \hat{w}(G_i)^{\uparrow}$). Choose a maximum cut of G_i . Then, the cut contains an edge with estimated length $\hat{w}(G_i)^{\uparrow} > 0$. We claim that every edge on this cut has a positive estimated length. This is proved as follows. Assume that the cut has an estimated length of 0. Since the cut is maximum, there exists a sub FJ-DAG of G_i with top and bottom edges in which all edges have an estimated length of 0 (the minimal case is a single vertex with top and bottom edges). Let $G_{(k)}$ be a maximal such sub FJ-DAG of G_i . $G_{(k)} \neq G_i$ since G_i has an edge with a positive estimated length. Then, there exists a sub FJ-DAG of G_i that shares top and bottom vertices with $G_{(k)}$ and has a positive length. This contradicts the assumption that F is homogeneous.

Decrement the estimated length of every edge on the cut by one. If e_i^t (e_i^b) is saturated and e_i^b (e_i^t) is unsaturated, then increment $\hat{w}(e_i^b)$ ($\hat{w}(e_i^t)$) by one. If both e_i^t and e_i^b are unsaturated, then increment $\hat{w}(e_i^t)$ or $\hat{w}(e_i^b)$ by one if $\hat{w}(e_i^t) = \hat{w}(e_i^b)$; increment $\hat{w}(e_i^t)$ by one if $\hat{w}(e_i^t) < \hat{w}(e_i^b)$; increment $\hat{w}(e_i^b)$ by one if $\hat{w}(e_i^t) > \hat{w}(e_i^b)$ (see Figure 6).

The following holds:

- The estimate is still homogeneous with the same length after the update;
- By repeating this procedure, C3 eventually becomes true in G_i ;
- If G_i has more than one children, then the characteristic vector decreases after the update. If G_i has only one child, then the characteristic vector decreases or does not change after the update;
- If C2 is true in a sub FJ-DAG $G_{(j)}$ of G , then C2 is still true in $G_{(j)}$ after the update. This is because the decrement is applied to edges on the maximum cut, and the increment is applied in such a way that C2 still holds;
- $\hat{w}(G_{(j)})^\uparrow$ may decrease in some FJ-DAG $G_{(j)}$ of G . This change does not make C3 false in any sub FJ-DAG.

Violation of C4: We assume C2 and C3 are true in any sub FJ-DAG. Let G_i be a sub FJ-DAG in which C4 is false. Then, $\hat{w}(e_i^t) > \hat{w}(G_i)^\downarrow + 1$ or $\hat{w}(e_i^b) > \hat{w}(G_i)^\downarrow + 1$ holds. Choose an unsaturated minimum cut of G_i . Increment every edge on the cut by one. Decrement e_i^t or e_i^b by one if $\hat{w}(e_i^t) = \hat{w}(e_i^b)$; decrement e_i^t by one if $\hat{w}(e_i^t) > \hat{w}(e_i^b)$; decrement e_i^b by one if $\hat{w}(e_i^t) < \hat{w}(e_i^b)$ (see Figure 7).

The following holds:

- The estimate is still homogeneous with the same length after the update;
- By repeating this procedure, C4 eventually becomes true in G_i ;
- The characteristic vector decreases after the update;
- If C2 is true in a sub FJ-DAG $G_{(j)}$ of G , then C2 is still true in $G_{(j)}$ after the update. This is because the increment is applied to edges on the minimum cut, and the decrement is applied in such a way that C2 still holds;
- $\hat{w}(G_{(j)})^\uparrow$ may decrease in some FJ-DAG $G_{(j)}$ of G . This change does not make C3 false in any sub FJ-DAG;
- $\hat{w}(G_{(j)})^\downarrow$ may increase in some FJ-DAG $G_{(j)}$ of G . This change does not make C4 false in any sub FJ-DAG;
- When $\hat{w}(G_i)^\uparrow = \hat{w}(G_i)^\downarrow$, $\hat{w}(G_i)^\uparrow$ increases and C3 may become false. We show that this does not happen. Since C3 is true in G_i before the update, $\hat{w}(e_i^t) \geq \hat{w}(G_i)^\uparrow$ ($\hat{w}(e_i^b) \geq \hat{w}(G_i)^\uparrow$) holds if e_i^t (e_i^b) is unsaturated. Suppose that e_i^t is updated. By the procedure, $\hat{w}(e_i^t) \geq \hat{w}(e_i^b)$ holds before the update. If e_i^b is saturated, then C3 is still true after the update. Suppose that e_i^b is unsaturated. Then, $\hat{w}(e_i^t) \leq \hat{w}(e_i^b) + 1$ holds by C2. Since $\hat{w}(e_i^t) > \hat{w}(G_i)^\downarrow + 1 = \hat{w}(G_i)^\uparrow + 1$ before the update, $\hat{w}(e_i^t) \geq \hat{w}(G_i)^\uparrow$ holds after the update. Moreover, $\hat{w}(e_i^b) + 1 \geq \hat{w}(e_i^t) > \hat{w}(G_i)^\uparrow + 1$ holds before the update, and therefore $\hat{w}(e_i^b) \geq \hat{w}(G_i)^\uparrow$ holds after the update. Hence, C3 is still true in G_i even when $\hat{w}(G_i)^\uparrow$ is incremented by one. This also holds when e_i^b is updated.

By applying these procedures, we eventually obtain a canonical estimate F^* with the same length such that $\#_{F^*} \preceq \#_F$. \square

By Proposition 1 and Lemma 3, we have the following theorem that characterizes the optimal estimates.

Theorem 1. *The characteristic vector of any canonical estimate with length h is minimum with regard to \preceq in all homogeneous estimates with length h .*

Using Lemma 1, we have the following corollary.

Corollary 1. *Let F is a terminal estimate for an FJ-DAG G . If F is canonical, then it is optimal.*

5. A Solution to PFJUEL

In this section, we present a solution method to PFJUEL that gives a canonical estimate for FJ-DAGs. It is a strategy defined as follows Algorithm 1:

Algorithm 1: Shortest Path Increment from Outmost Edges (SPIOE):

```

1: While all goal paths have at least one unsaturated edge:
2:   Let  $S$  be the set of such an unsaturated edge  $e$  that
3:   (i)  $e$  is on a goal path with the shortest estimated length, and
4:   (ii)  $e$  is not in any sub FJ-DAG for which the current
5:       estimate is terminal;
6:   If  $|S| > 1$ , then let  $S'$  be the set of edges in  $S$ 
7:       having the estimated length minimum in  $S$ ;
8:   else  $S' := S$ ;
9:   If  $|S'| > 1$ , then let  $S''$  be the set of edges in  $S'$ 
10:      having the level highest in  $S'$ ;
11:   else  $S'' := S'$ ;
12:   Choose one edge from  $S''$  and issue an oracle call to it;
13: endwhile.
```

Remark 1. At line 12, there may be more than one edge in S'' . To make the strategy deterministic, we need some rule to select one edge; e.g., introducing a total order to the set of edges.

Let h^* be the length of a shortest goal path. Suppose that an edge (v_i, v_j) is selected in the while loop, and (v_i, v_j) is on a goal path π whose estimated length already reaches h^* . If $\hat{w}(v_i, v_{(i+1)}) < w(v_i, v_{(i+1)})$, then $search_c(v_i, v_{(i+1)})$ returns “no” and the estimated length of π becomes $h^* + 1$. Such an oracle call is unnecessary for satisfying the requirement R2. We call such a fail call a waste fail call. Waste fail calls are unavoidable if the length of each edge is initially unknown.

Lemma 4. If waste fail calls do not occur, then the estimate given by SPIOE is canonical.

Proof. If the oracle call is not a success call, then SPIOE increments the length of a goal path with the shortest estimated length by one. Since waste fail calls do not occur, the estimated length of every goal path does not exceed the length of the shortest goal path, and therefore SPIOE gives a homogeneous estimate when it terminates.

We show that the violation of C2–C4 does not occur in any sub FJ-DAG during the execution of SPIOE. This is proved by mathematical induction in the number of iterations. Initially, the estimate is canonical.

Assume that the current estimate satisfies C2–C4 in any sub FJ-DAG. Suppose that the estimated length of the upper edge e_i^t of a sub FJ-DAG G_i has been updated; i.e., incremented by one.

We first show that C2–C4 still hold in G_i after the update.

C2: If e_i^b is saturated just before the update, then a violation of C2 does not occur. Suppose that e_i^b is unsaturated just before the update. Note that every goal path that contains e_i^t also contains e_i^b . Then, by line 7, $\hat{w}(e_i^t) \leq \hat{w}(e_i^b)$ holds just before the update. Therefore, C2 is still true after the update.

C3: Since $\hat{w}(e_i^t) \geq \hat{w}(G_i)^\uparrow$ holds just before the update, then C3 is still true after the update.

C4: If $\hat{w}(e_i^t) < \hat{w}(G_i)^\downarrow + 1$ just before the update, then C4 is still true after the update. Suppose that $\hat{w}(e_i^t) = \hat{w}(G_i)^\downarrow + 1$ just before the update. Any goal path that contains e_i^t also contains an edge in a minimum unsaturated cut of G_i . Then, the path contains an unsaturated edge with an estimated length no greater than $\hat{w}(G_i)^\downarrow$. By line 7, SPIOE does not choose e_i^t because $\hat{w}(e_i^t)$ is not minimum in unsaturated edges on the path.

The case that e_i^b has been updated is similar. Next, we show that C2–C4 are still true in other sub FJ-DAGs after the update. Since C2 is the condition for the top and the bottom edge of each sub FJ-DAG, C2 is still true in any sub FJ-DAG after the update. In any sub FJ-DAG that does not contain e_i^t , C3 and C4 are still true after the update. Thus, we need to check the conditions for sub FJ-DAGs that contain e_i^t . Let $G_{(j)}$ be such a sub FJ-DAG. Then, $\hat{w}(G_{(j)})^\uparrow$ and $\hat{w}(G_{(j)})^\downarrow$ may be incremented by one. Suppose that $\hat{w}(G_{(j)})^\uparrow$ is incremented. This occurs when $\hat{w}(e_i^t) = \hat{w}(G_{(j)})^\uparrow$ just before the update. C3 becomes false after the update only if for the top or bottom edge $e_{(j)}^x$ of $G_{(j)}$, $\hat{w}(e_{(j)}^x) = \hat{w}(e_i^t)$ holds just before the update. Every goal path that contains e_i^t also contains $e_{(j)}^x$ because $e_{(j)}^x$ is in a level higher than that of e_i^t . Therefore, SPIOE does not choose e_i^t by line 10, and this case does not occur. Clearly, C4 is still true when $\hat{w}(G_{(j)})^\downarrow$ is incremented. Thus, C3 and C4 are still true in other sub FJ-DAGs.

By the above arguments, C2–C4 are still true in all sub FJ-DAGs after the update. Hence, we can conclude that the obtained estimate is canonical. \square

Remark 2. The condition at line 4–5 is not used in the above proof. Dropping this condition may cause waste fail calls and unnecessary success calls.

From Corollary 1 and Lemma 4, we obtain the main theorem.

Theorem 2. If waste fail calls do not occur, then SPIOE always gives an optimal estimate.

The detailed complexity analysis of the SPIOE is omitted for the following reasons. The analysis can be done for (i) the characteristic vector of the optimal terminal estimate obtained by SPIOE and (ii) the total computational time of SPIOE. The optimal terminal estimate is characterized by Corollary 1 (i.e., canonical estimate) and is determined by the given FJ-DAG. It may be possible to give trivial lower/upper bounds of the characteristic vector, but they are meaningless. The total computational time of SPIOE is obtained by evaluating operations in each iteration and the total number of iterations. Operations in each iteration consist of maintaining shortest goal paths and finding an edge with the minimum estimated length in the highest level. They can be done in polynomial time. Furthermore, we can give an upper bound of the number of iterations, which is determined by the optimal terminal estimates. This is apparently polynomial in the size of the graph and edge length. Such detailed complexity analysis is not very valuable under the above cost assumption on calling oracles. Moreover, the goal of the paper is to develop an optimal strategy that can be applied to real problems. In this sense, the goal has been achieved.

Finally, we analyze the number of waste fail calls. Let h^* be the length of the shortest goal path. We introduce a special class of strategies. Strategies that always issue oracle calls to unsaturated goal paths with the shortest estimated length are called rational. In the terminal estimate obtained by any rational strategy, the estimated length of every goal path is always less than or equal to $h^* + 1$, and therefore the number of waste fail calls is less than the number of goal paths.

Lemma 5. Let g be the number of goal paths. Then, $g - 1$ is an upper bound of the number of waste fail calls by any rational strategy.

For a strategy that is not rational, the estimated length of some goal path may become greater than $h^* + 1$. Suppose that the estimated length of the current shortest goal path has reached h^* and the strategy issues an oracle call to a goal path with length $h^* + 1$. We can make an instance of FJ-DAGs for which the oracle call fails and the estimated length becomes $h^* + 2$.

There exists an FJ-DAG for which any rational strategy gives $g - 1$ waste fail calls; i.e., $g - 1$ is the tight upper bound for rational strategies. Let us consider an FJ-DAG shown in Figure 9, where every edge has length 1. At each round, any rational strategy picks up one

unsaturated edge on a goal path with the shortest estimated length and issues an oracle call to it. Then, we will eventually reach the following situation: all edges are saturated but are in E_U . The next oracle call is issued to some edge and it succeeds. Then, we consider an instance having a length of 2 for this edge. The strategy shows the same behavior for the new instance before this oracle call, but now the oracle call fails. Since the strategy is rational, the next oracle call is issued to an edge on a goal path with an estimated length of 2. We can repeat this $g - 1$ times. After that, we obtain a goal path with a length of 2 such that all edges on it are in E_K . When a terminal estimate is obtained, the number of waste fail calls issued so far is $g - 1$. Hence, we have the following result.

Lemma 6. *Given any rational strategy, there is an instance of FJ-DAGs for which the the number of waste fail calls is $g - 1$, where g is the number of goal paths.*

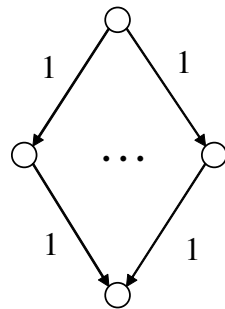


Figure 9. An FJ-DAG that gives $g - 1$ waste fail calls.

We demonstrate the execution of strategy SPIOE. Consider an FJ-DAG in Figure 10a. The obtained terminal estimate by SPIOE is shown in Figure 10b. Bold arrows indicate that success calls occur at the edges. The shortest goal path is $(1, 2, 3, 6, 7)$.

The execution trace is shown in Table 1, where each column indicates the estimated length of the current shortest goal path, the number c and the edge (v_i, v_j) in oracle call $serach_c(v_i, v_j)$, and the result. In this trace, no waste fail calls occur.

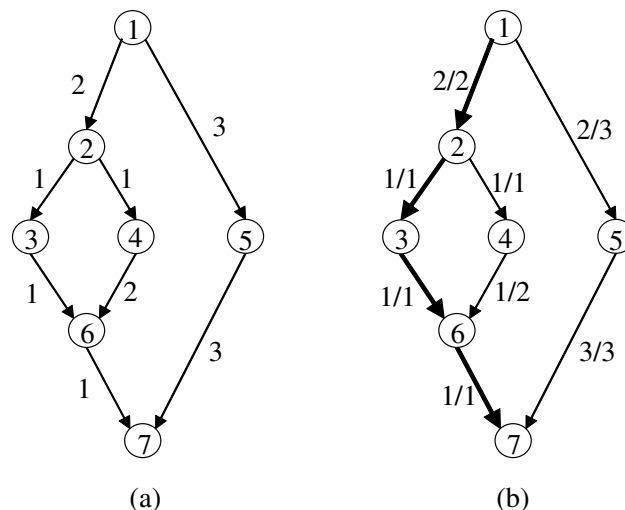


Figure 10. (a) An FJ-DAG and (b) an optimal estimate.

Table 1. An execution trace of SPIOE.

| Step | Length | c | Edge | Result |
|------|--------|-----|--------|--------|
| 1 | 0 | 0 | (6, 7) | no |
| 2 | 0 | 0 | (1, 5) | no |
| 3 | 1 | 0 | (1, 2) | no |
| 4 | 1 | 0 | (5, 7) | no |
| 5 | 2 | 0 | (2, 3) | no |
| 6 | 2 | 0 | (2, 4) | no |
| 7 | 2 | 1 | (5, 7) | no |
| 8 | 3 | 0 | (3, 6) | no |
| 9 | 3 | 0 | (4, 6) | no |
| 10 | 3 | 1 | (1, 5) | no |
| 11 | 4 | 1 | (1, 2) | no |
| 12 | 4 | 2 | (5, 7) | no |
| 13 | 5 | 1 | (6, 7) | yes |
| 14 | 5 | 1 | (3, 6) | yes |
| 15 | 5 | 1 | (2, 3) | yes |
| 16 | 5 | 2 | (1, 2) | yes |

6. Conclusions

We have studied a pathfinding problem of FJ-DAGs with unknown edge length and have proposed a strategy that gives a shortest path. The strategy is optimal in the sense that the characteristic vector consisting of the number of fail oracle calls for each length is minimized with respect to a lexicographical order, provided that there are no waste fail calls. Theoretical upper bounds on the number of waste fail calls are also given. As a result, several technical problems that remained in the previous paper have been solved; i.e., showing optimality for a stronger criterion, which was initially intended to use in the previous paper, and making the discussion clearer.

Although PFJUEL is a purely theoretical problem, solution methods for it can contribute to reducing the cost of configuration change in reconfigurable cloud computing systems. The proposed strategy is an optimal one under a reasonable assumption; i.e., no other strategy is better than this. This is a notable result. Extending the targets to graphs with cycles remains as future work. In real applications, if the upper bound of the edge length is known, then the average computational cost may be reduced by non-conservative strategies. This issue should be also studied in the future.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Hiraishi, K.; Kobayashi, K. A Pathfinding Problem for Search Trees with Unknown Edge Length. *J. Discret. Algorithms* **2018**, *49*, 1–7. [\[CrossRef\]](#)
2. Kikuchi, S.; Tsuchiya, S. Configuration Procedure Synthesis for Complex Systems Using Model Finder. In Proceedings of the 15th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2010), Oxford, UK, 22–26 March 2010; pp. 95–104.
3. Kikuchi, S.; Tsuchiya, S.; Hiraishi, K. Synthesis of Configuration Change Procedure Using Model Finder. *IEICE Trans. Inf. Syst.* **2013**, *E-96-D*, 1696–1706. [\[CrossRef\]](#)
4. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [\[CrossRef\]](#)
5. Korf, R. Depth-first Iterative-Deepening: An Optimal Admissible Tree Search. *Artif. Intell.* **1985**, *27*, 97–109. [\[CrossRef\]](#)

6. Björnsson, Y.; Enzenberger, M.; Holte, R.C.; Schaeffer, J. Fringe Search: Beating A^* at Pathfinding on Game Maps. In Proceedings of the 2005 IEEE Symposium on Computational Intelligence and Games, Colchester, Essex, UK, 4–6 April 2005; pp. 125–132.
7. Russell, S. Efficient Memory-bounded Search Methods. In Proceedings of the 10th European Conference on Artificial Intelligence, Vienna, Austria, 3–7 August 1992; pp. 1–5.
8. Nikolova, E.; Kelner, J.A.; Brand, M.; Mitzenmacher, M. Stochastic Shortest Paths via Quasi-convex Maximization. *Theor. Comput. Sci.* **2006**, *4168*, 552–563.
9. Papadimitriou, C.H.; Yannakakis, M. Shortest Paths Without a Map. In *Lecture Notes in Computer Science*; (Proc. 16th ICALP); Springer: Berlin/Heidelberg, Germany, 1989; Volume 372, pp. 610–620.
10. Karger, D.; Nikolova, E. *Exact Algorithms for the Canadian Traveler Problem on Paths and Trees*; MIT CSAIL Technical Report; MIT: Cambridge, MA, USA 2008; MIT-CSAIL-TR-2008-004.
11. Gyögy, A.; Linder, T.; Lugosi, G.; Ottucsák, G. The On-Line Shortest Path Problem Under Partial Monitoring. *J. Mach. Learn. Res.* **2007**, *8*, 2369–2403.
12. Khani, A. An Online Shortest Path Algorithm for Reliable Routing in Schedule-based Transit Networks Considering Transfer Failure Probability. *Transp. Res. Part B* **2019**, *126*, 549–564. [[CrossRef](#)]
13. Blei, D.; Kaelbling, L. Shortest Paths in a Dynamic Uncertain Domain. In Proceedings of the IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments, Stockholm, Sweden, 31 July–2 August 1999.
14. Boyan, J.; Mitzenmacher, M. Improved Results for Route Planning in Stochastic Transportation Networks. In Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms, Washington, DC, USA, 7–9 July 2001; pp. 895–902.
15. Nikolova, E.; Brand, M.; Karger, D.R. Optimal Route Planning Under Uncertainty. In Proceedings of the International Conference on Automated Planning and Scheduling, Cumbria, UK, 6–10 June 2006; pp. 131–140.
16. Karp, R.M. On-Line Algorithms Versus Off-Line Algorithms: How Much is it Worth to Know the Future? In Proceedings of the IFIP 12th World Computer Congress on Algorithms, Software, Architecture-Information Processing '92, Madrid, Spain, 7–11 September 1992; Volume 1, pp. 416–429.
17. Baccelli, F.; Massey, W.A.; Towsley, A. Acyclic Fork-join Queuing Networks. *J. ACM* **1989**, *36*, 615–642. [[CrossRef](#)]