On fault-tolerant path optimization under QoS constraint in multi-channel wireless networks

Shurong Zhang, Lin Chen, Weihua Yang

A College of Mathematics, Taiyuan University of Technology, Taiyuan, Shanxi 030024, China
B Lab. Recherche Informatique (LRI-CNRS UMR 8623), Univ. Paris-Sud, 91405 Orsay, France

A R T I C L E   I N F O

Article history:
Received 4 August 2016
Received in revised form 15 June 2017
Accepted 29 July 2017
Available online 2 August 2017
Communicated by C. Kaklamanis

Keywords:
Multi-channel wireless networks
Vertex- and channel-disjoint paths
Fully polynomial-time approximation algorithms

A B S T R A C T

In multi-channel wireless networks, a fundamental problem is to find node-disjoint paths minimising global cost or the maximum individual path cost, under the constraint that each path operates on a separate channel to maximise the fault tolerance and robustness against channel instability and malicious attacks. Meanwhile, the quality of service (QoS) requirement (e.g., in terms of end-to-end delay) needs to be satisfied on each path. In this paper, we provide a comprehensive formulation and analysis on this multi-path optimization problem by casting it to the problem k-disjoint path with different colours (k-DPDC). We further formulate the Restricted MinSum k-DPDC and Restricted MinMax k-DPDC to denote the problems of finding multiple node- and channel-disjoint paths minimising the global cost and the maximum individual path cost under the QoS constraint on the path end-to-end delay. Given the NP-hardness of both problems, we focus on directed acyclic graphs and propose fully polynomial-time approximation algorithms for both problems.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

We consider the following fundamental path optimization problem arising from reliable and fault-tolerant routing in wireless communication networks. Given a wireless network composed of n nodes, each of which operates on one of λ orthogonal frequency channels (indexed from 1 to λ), except for some nodes containing the source node s and the destination node t which can access all the λ channels, we seek k ≤ λ node-disjoint paths from s to t satisfying the following constraints:

- Fault tolerance. We call a path P, channel c path, if P is composed of only nodes which can operate on channel c. Two paths P1 and P2 are channel-disjoint if there exist distinct c1 and c2 with 1 ≤ c1, c2 ≤ λ such that P1 (i = 1, 2) is a channel c1 path. As the first constraint, we require that any pair of the k paths are channel-disjoint one to the other. From the engineering perspective, this constraint implies that as long as one channel is operational, we can ensure that the packets can arrive at the destination t, i.e., the k paths we seek can tolerate up to k − 1 channels problem, e.g., blocked by attackers.

* This work is supported by the National Natural Science Foundation of China (61502330, 11671296). Part of the work presented in the paper is done while S. Zhang was visiting LRI, Univ. Paris-Sud.
* Corresponding author.
E-mail address: shurongmm@163.com (S. Zhang).

http://dx.doi.org/10.1016/j.tcs.2017.07.027
0304-3975/© 2017 Elsevier B.V. All rights reserved.
Quality-of-service constraint. We require that the delay induced by each path is upper-bounded by a threshold $D$.

Performance optimization. We require that the global performance of the $k$ paths is optimal. Specifically, we consider two natural performance metrics widely used in wireless routing problems, (1) the sum of cost (e.g., in terms of energy consumption) of all the $k$ paths and (2) the maximum cost of the $k$ paths, both of which are explored in our study.

We emphasise that it is the combination of the above design requirements that makes our path optimization problem far from trivial and should be handled holistically.

2. Problem formulation

In this section, we formulate the path optimization problem posed in Introduction. We first formulate the $k$-disjoint path with different colours ($k$-DPDC) problem by casting channels to colours in which we seek $k$ channel-disjoint paths. We then incorporate the constraints on fault tolerance and quality of service [8,13–15] under the two performance metrics identified in Introduction.

Given a wireless network and $\lambda$ channels $1, 2, \ldots, \lambda$, assume that each node $v$ (1) operates on only one channel $i \in [1, \lambda]$ or (2) operates on all channels. In cases (1) and (2), $v$ is assigned the colour $i$ and colour 0, respectively. So, the main problems can be defined in a general graph $G$ and every node of $G$ has a colour $i \in [0, \lambda]$. In this paper, we will consider the case that $G$ is a directed acyclic graph (DAG). For a graph $G$ with vertex set $V(G)$ and edge set $E(G)$, a partial $\lambda$-vertex colouring of $G$ is a mapping $\text{col} : V(G) \rightarrow \{0, 1, \ldots, \lambda\}$ such that there exist at least two vertices of the colour 0. If $\text{col}(v) = 0$ for every $v \in V(G)$, the partial $\lambda$-vertex colouring of $G$ is called an empty $\lambda$-colouring. Let $P = \{v_1, v_2, \ldots, v_l\}$ be a path in $G$. If $\text{col}(v_i) = 0$ for $i = 0, 1, \ldots, l$, set $\text{col}(P) = 0$; if the colour of some vertex in $P$ is not 0 and there exists a colour $c \in [1, \lambda]$ such that $\text{col}(v_i) \in \{0, c\}$ for $i = 0, 1, \ldots, l$, set $\text{col}(P) = c$.

**Definition 2.1** ($k$-DPDC problem). Given a graph $G = (V(G), E(G))$ with a partial $\lambda$-vertex colouring such that both the source vertex $s$ and target vertex $t$ have colour 0. Let $k \leq \lambda$ be a constant representing the number of vertex-disjoint paths to be found from $s$ to $t$. The problem is to find $k$-vertex-disjoint $st$-paths $P_1, P_2, \ldots, P_k$ such that for each path $P_i$, $\text{col}(P_i)$ exists and for any two different paths $P_i$ and $P_j$, we must have $\text{col}(P_i) \neq \text{col}(P_j)$ or $\text{col}(P_i) = \text{col}(P_j) = 0$.

We denote the $k$ pairwise internally disjoint $st$-paths $P_1, P_2, \ldots, P_k$ as required in Definition 2.1 by $k$ vertex-disjoint $st$-paths with different colours. Obviously, if $G$ is a graph with an empty $\lambda$-colouring, then the $k$-DPDC problem is same as the $k$ vertex-disjoint paths problem.

In the following, in order to seek $k$ vertex-disjoint $st$-paths with different colours minimising the total and maximum path cost while satisfying the QoS constraint, we formulate the restricted MinSum $k$-DPDC problem and the restricted MinMax $k$-DPDC problem.

**Definition 2.2** (Restricted MinSum $k$-DPDC problem). Given a graph $G$ with a partial $\lambda$-vertex colouring, two distinct vertices $s, t \in V(G)$ with colour 0, a positive integer $D$, a cost function $c : E(G) \rightarrow \mathbb{R}^+$ and a delay function $d : E(G) \rightarrow \mathbb{R}^+$, where $\mathbb{R}^+$ is a set of positive real numbers, we want to find $k$ vertex-disjoint $st$-paths $P_1, P_2, \ldots, P_k$ with different colours such that $\sum_{i=1}^{k} c(P_i)$ is minimised and $\sum_{i=1}^{k} d(P_i) \leq D$, where $c(P_i) = \sum_{e \in E(P_i)} c(e)$ and $d(P_i)$ is defined similarly.

When $k = 1$ and $G$ is an graph with an empty $\lambda$-colouring, the restricted MinSum $k$-DPDC problem is the same as the restricted shortest path problem which is NP-complete [6]. Further assuming that $G$ is a directed acyclic graph (DAG), Warburton proposed a polynomial approximation scheme firstly [18]. Then, using a technique called rounding-and-scaling [16], two simple fully polynomial approximation schemes (FPASs) were given in [11] and [3], respectively. We will give an FPAS for restricted MinSum $k$-DPDC problem ($k \geq 2$) on a DAG in Section 4 of this paper.

**Definition 2.3** (Restricted MinMax $k$-DPDC problem). Given a graph $G$ with a partial $\lambda$-vertex colouring, two distinct vertices $s, t \in V(G)$ with colour 0, a positive integer $D$, a cost function $c : E(G) \rightarrow \mathbb{R}^+$ and a delay function $d : E(G) \rightarrow \mathbb{R}^+$, we want to find $k$ vertex-disjoint $st$-paths $P_1, P_2, \ldots, P_k$ with different colours such that $\max_{1 \leq i \leq k} c(P_i)$ is minimised and $\sum_{i=1}^{k} d(P_i) \leq D$.

When the partial $\lambda$-colouring of $G$ is empty and the delay constraint is infinity, the restricted MinMax $k$-DPDC problem is the MinMax $k$-disjoint paths (MinMax $k$-DP) problem proposed by Li et al. [10]. They proved that this problem is strongly NP-complete when $k = 2$. Further more, suppose that $G$ is a DAG, Fleischer et al. developed an FPTAS for this problem [4]. An improved approximation scheme was given by Yu et al. [20]. Based on this, Wu gave a faster approximation scheme [19]. In Section 5, we investigate the restricted MinMax $k$-DPDC problem on a DAG and present an FPTAS.

3. Technical preliminaries

In this section, we give some useful notations firstly and then an important auxiliary graph proposed by Li et al. [10] will be introduced. It is natural that a wireless network may be modelled by a directed graph $G$ consisting of the vertex set
Lemma 3.1 ([20]). We can construct $G_k$ in $O(n^{k+1})$ time such that $|V_k| = O(n^k)$ and $|E_k| = O(n^{k+1})$.

Any edge $e = ((v_1, \ldots, v_d), (v_1, v_{d+1}, \ldots, v_k)) \in E_k$ is called a $d$-dimensional edge and edge $(v_d, v_d')$ is the corresponding edge in $G$. In addition, $e$ is associated with a delay $d_0(e) = d((v_d, v'_d))$ and $k + 1$ costs $c_0(e), c_1(e), c_2(e), \ldots, c_k(e)$, where $c_0(e) = c((v_d, v'_d))$ and for every $i \in [1, k]$,

$$c_i(e) = \begin{cases} c((v_d, v'_d)), & \text{if } i = d; \\ 0, & \text{if } i \neq d. \end{cases}$$

Then, using the following property (Lemma 3.2), the $k$ vertex-disjoint $st$-paths problem on $G$ with different kinds of restrictions can be reduced to a multi-weighted $s^*t^*$-path problem on $G_k$.

Lemma 3.2 ([20]). There are $k$ pairwise internally disjoint $st$-paths $P_1, P_2, \ldots, P_k$ in $G$ if and only if there is an $s^*t^*$-path $P$ in $G_k$ and $P_i$ consists of all the edges which correspond to $i$-dimensional edges of $P$.

**Lemma 3.2** implies that $c_0(P) = \sum_{i=1}^k c(P_i)$, $c_i(P) = c(P_i)$ for $i = 1, 2, \ldots, k$ and $d_0(P) = \sum_{i=1}^k d(P_i)$. Recall that there is no directed cycle in $G$. Then, an important lemma about $G_k$ is given as follows.

**Lemma 3.3.** $G_k$ is a DAG.

**Proof.** For simplicity, we only prove the lemma for the case $k = 2$. By contradiction. Let $C = \langle (u_1, v_1), (u_2, v_1), \ldots, (u_i, v_j), (u_j, v_{j+1}), \ldots, (u_1, v_1) \rangle$ be a directed cycle in $G_2$. Then for any $i \in [1, 2]$, all the corresponding edges of $i$-dimensional edges compose a subgraph $C_i$ in $G$, where $C_1 = \langle (u_1, u_2), \ldots, (u_i, u_1) \rangle$ and $C_2 = \langle (u_1, v_j), \ldots, (u_1, v_1) \rangle$. Obviously, $|N^+_C(w)| = |N^+_C(w)|$ for every $w \in V(C_i)$ and so $C_i$ contains a directed cycle for $i = 1, 2$, a contradiction to the fact that $G$ is a DAG. □

Since $G_k$ is a DAG, it is possible to find a topological ordering $w_1, w_2, \ldots, w_{|V_k|}$ of the vertices in $V_k$, i.e. $i < j$ for every $(w_i, w_j) \in E_k$.

For any vertex $v = \langle v_1, v_2, \ldots, v_k \rangle \in V_k$ and $i \in [1, k]$, set $col_i(v) = col(v_i)$. Let $P$ be a path in $G_k$. The $i$-dimensional path $P_i$ of $P$ consists of all the edges corresponding to $i$-dimensional edges of $P$. If $col(P_i)$ exists, then $col(P_i)$ is said to be the $i$-dimensional colour of $P$, denoted by $col_i(P)$. When $col_i(P)$ exists for each $i \in [1, k]$, we say that $P$ has property $P$ if for any different indices $i, j \in [1, k]$, $col_i(P) \neq col_j(P)$ or $col_i(P) = col_j(P) = 0$. Then following corollary of **Lemma 3.2** holds obviously.

**Corollary 3.4.** There are $k$-disjoint $st$-paths $P_1, P_2, \ldots, P_k$ with different colours in $G$ if and only if there is an $s^*t^*$-path $P$ with property $P$ in $G_k$ and $P_i (i = 1, 2, \ldots, k)$ is the $i$-dimensional path of $P$.

4. The restricted MinSum $k$-DPDC problem

Denote by $OPT(G)$ the total cost of an optimal solution for this problem. In Section 4.1, for the special case that $c(e) \in \mathbb{N}^+$ for each $e \in E(G)$, where $\mathbb{N}^+$ is the set of non-negative integer, we propose an algorithm to compute $OPT(G)$. Then, using this algorithm and the technique named rounding-and-scaling [5,7,9,17], an FPTAS for the restricted MinSum $k$-DPDC problem in a general case $c(e) \in \mathbb{R}^+$ is presented in Section 4.2.
4.1. Restricted MinSum k-DPDC problem when \( c(e) \in \mathbb{N}^+ \)

In this section, we suppose that \( c(e) \in \mathbb{N}^+ \) for each \( e \in E(G) \). By Corollary 3.4, in order to obtain \( \text{OPT}(G) \) of the restricted MinSum DPDC problem, we need to find an \( s^*t^* \)-path \( P \) with property \( \mathcal{P} \) in \( G_k \) such that \( c_0(P) \) is minimal and \( d_0(P) \leq D \). Recall that \( w_1, w_2, \ldots, w_{|V_k|} \) is a topological ordering of the vertices in \( V_k \) such that \( i < j \) for every \( (w_i, w_j) \in E_k \). For clarity of presentation, we assume that \( w_1 = s^* \) and \( w_{|V_k|} = t^* \). We now list the main steps.

Step 1. Given an integer \( C \), for any \( w_i \in V_k \), let \( M_{w_i}[C] \) be the minimal \( d_0 \)-weight of \( s^*w_i \)-path \( P \) which satisfies property \( \mathcal{P} \) and the constraint \( c_0(P) \leq C \). We will compute \( M_{w_i}[C] \) in this step.

Step 2. Note that \( \min(C : M_{w_i}[C] \leq D) \) is the optimal solution \( \text{OPT}(G) \) for the restricted MinSum k-DPDC problem when \( c(e) \in \mathbb{N}^+ \) for each \( e \in E(G) \).

4.1.1. The computation of \( M_{w_i}[C] \)

For any \( w_i \in V_k \), let \( C_{w_i} \) be a subset of \( U = \{(x_1, x_2, \ldots, x_k) : 1 \leq x_0 \neq x_b \leq \lambda \text{ for any } a \neq b\} \) such that \( (x_1, x_2, \ldots, x_k) \in C_{w_i} \) if and only if there exists an \( s^*w_i \)-path \( P \) in \( G_k \) such that \( c_0(P) = x_a \) when \( c_0(P) \neq 0 \) for each \( \alpha \in [1, k] \).

According to the above description, if \( X = (x_1, x_2, \ldots, x_k) \in C_{w_i} \) where \( w_j \in V_k \) and \( N^+_k(w_j) \neq \emptyset \), then there is an \( s^*w_j \)-path \( P \) in \( G_k \) such that \( c_0(P) = x_a \) when \( c_0(P) \neq 0 \) for each \( \alpha \in [1, k] \). Choose \( w_i \in N^+_k(w_j) \). Suppose that \( c_0(P) = x_a \) when \( c_0(P) \neq 0 \) for each \( \alpha \in [1, k] \). Thus \( P' = P + (w_j, w_i) \) is an \( s^*w_i \)-path and \( c_0(P') = x_a \) when \( c_0(P') \neq 0 \) for each \( \alpha \in [1, k] \). Obviously, \( X \in C_{w_i} \) and \( P' \) satisfies property \( \mathcal{P} \). Hence, we have that if \( C = \bigcup_{w_j \in N^+_k(w_j)} C_{w_j} \), then

\[
C_{w_i} = \{x_1, x_2, \ldots, x_k \in C : c_0(w_i) = x_a \text{ when } c_0(w_i) \neq 0 \text{ for each } \alpha \in [1, k] \}.
\]

So, in order to construct the \( s^*w_i \)-path with property \( \mathcal{P} \) in \( G_k \), it is enough to consider all the elements in \( C_{w_i} \). Suppose that \( C_{w_i} \neq \emptyset \) and \( i \geq 2 \). Choose \( X = (x_1, x_2, \ldots, x_k) \in C_{w_i} \). If there exists an \( s^*w_i \)-path \( P \) with property \( \mathcal{P} \) such that \( c_0(P) = x_a \) and \( c_0(P) = x_a \) when \( c_0(P) \neq 0 \) for each \( \alpha \in [1, k] \), the minimal \( d_0 \)-weight of \( P \) is denoted by \( T_{w_i}[C, X] \); otherwise, let \( T_{w_i}[C, X] = \infty \). Note that \( w_1 = s^* \). Set \( T_{w_i}[0, X] = 0 \).

Now, if we may choose a nonempty vertex subset \( N_{w_i} = \{w_j \in N^+_k(w_i) : j \geq 1, C - c_0((w_j, w_i)) \geq 0 \} \), we can observe that the following equation holds.

\[
T_{w_i}[C, X] = \min_{w_j \in N_{w_i}} \{T_{w_j}[C - c_0((w_j, w_i)), X] + d_0((w_j, w_i))\}.
\]

Based on this, if we let \( T_{w_i}[C] = \min_{X \in C_{w_i}} \{T_{w_i}[C, X]\} \), then

\[
M_{w_i}[C] = \min_{1 \leq \lambda \leq C} \{T_{w_i}[\lambda]\} = \min \{M_{w_i}[C - 1], T_{w_i}[C]\}.
\]

Above computations of \( T_{w_i}[C, X] \) and \( M_{w_i}[C] \) are described in Procedure 1.

4.1.2. The computation of \( \text{OPT}(G) \)

When \( c(e) \in \mathbb{N}^+ \) for each \( e \in E_k \), if there exists an \( s^*t^* \)-path \( P \) with property \( \mathcal{P} \) such that \( c_0(P) \) is minimised and \( d_0(P) \leq D \), then it remains to compute \( \text{OPT}(G) = \min(C : M_{s^*}[C] \leq D) \). Therefore, begin with \( C = 1 \). If \( M_{t^*}[C] \leq D \), then \( \text{OPT}(G) = C \) and we can obtain the corresponding \( s^*t^* \)-path; otherwise, set \( C := C + 1 \) and proceed the next iteration. Then we propose Algorithm 1 as follows.

By Lemma 3.1, the construction of \( G_k = (V_k, E_k) \) in step 1 needs \( O(n^{k+1}) \) time. Line 2 takes \( O(|E_k|) \) time to obtain a topological order of \( V_k \). Clearly, Lines 3–4 take \( O(|E_k|) \) time. Since times \( O(N^+_k(w_i)) \) and \( O(|E_k|) \) are needed in Lines 6
Algorithm 1 Restricted MinSum $k$-DPDC problem when $c(e) \in \mathbb{N}^+$.

**Input:** A DAG $G$ with a partial $\lambda$-colouring and weight functions $c : E(G) \rightarrow \mathbb{N}^+$ and $d : E(G) \rightarrow \mathbb{R}^+$, two vertices $s$, $t$ with colour 0 and upper bound $D$.

**Output:** The optimal solution.

1. construct $G_k = (V_k, E_k)$ with weight functions $c_0, d_0$ and $col_0$ for each $\alpha \in [1, k]$;
2. find the topological ordering $w_1, w_2, \ldots, w_{|V_k|}$ of the vertices in $V_k$;
3. let $C = \{x_1, x_2, \ldots, x_\lambda : 1 \leq x_\alpha \neq x_b \leq \lambda \}$ for any $a \neq b$ and there exists $w \in N^\alpha_{G_k}(s^\alpha)$ such that $col_\alpha(w)$ when $col_\alpha(w) \neq 0$ for each $\alpha \in \{1, k\}$;
4. $T_{tr}[0, X] := 0$ for each $X \subseteq C^*$;
5. for $i \leftarrow 2$ to $|V_k|$ do
6. $C := \bigcup_{w_j \in N^\alpha_{G_k}(w_i)} C_{w_i}$;
7. $C_{w_i} := \{x_1, x_2, \ldots, x_\lambda \in C : x_\alpha = col_\alpha(w_i)$ when $col_\alpha(w_i) \neq 0$
               for each $\alpha \in \{1, \ldots, k\}$
8. $T_{tr}[0, X] := \infty$ for each $X \subseteq C_{w_i}$;
9. $M_{w_j}[0] := \infty$;
end for
10. $C := 1$;
11. for $i \leftarrow 2$ to $|V_k|$ do
12. use Procedure 1 to compute $M_{w_i}[C]$;
13. if $w_i = t^*$ and $M_{w_i}[C] \leq D$ do
14. return $OPT(G) = C$ and its corresponding paths, exit;
15. end if
16. end for
17. $C := C + 1$ and return to step 12.

and 7–8, respectively, the time complexity of the for loop in Lines 5–10 is $O\left(\binom{n}{k}|E_k| + |V_k|\right) = O\left(\binom{n}{k}n^{k+1}\right)$. We can observe that Algorithm 1 stops when $C \subseteq OPT(G)$, there are $OPT(G)$ iterations of Lines 12–17. For a given $C \in \{1, \ldots, OPT(G)\}$ and a fixed vertex $w_i(i \in [2, |V_k|])$, we can observe that Procedure 1 needs $O\left(\binom{n}{k}|N^\alpha_{G_k}(w_i)|\right)$ time and so the overall running time of Lines 12–17 is $\sum_{i=2}^{\lfloor 2/|V_k| \rfloor} O\left(\binom{n}{k}|N^\alpha_{G_k}(w_i)|\right) = O\left(\binom{n}{k}|E_k|\right) = O\left(\binom{n}{k}n^{k+1}\right)$. Therefore, the total time for Lines 12–18 is $O\left(\binom{n}{k}n^{k+1} OPT(G)\right)$ and so we have following result.

**Theorem 4.1.** For any fixed $\lambda \geq k > 1$, when $c(e) \in \mathbb{N}^+$, the Restricted MinSum $k$-DPDC problem on a directed acyclic graph can be solved in $O\left(\binom{n}{k}n^{k+1} OPT(G)\right)$ time.

4.2. The FPTAS

In this section, based on Algorithm 1 and the technique named rounding-and-scaling [3,5,7,9,11,16,17,19], we consider the efficient approximation for the restricted MinSum $k$-DPDC problem on a DAG $G$ in the general case that $c(e), d(e) \in \mathbb{R}^+$ for each $e \in E_k$. Let $G_k$ be a graph obtained from $G_k$ by deleting the edge set $\{e \in E_k : c(e) > 1\}$. If we replace weight $c(e)$ of any $e \in E(G)$ with $c_k/e$, we obtain a new weighted graph, denoted by $G_k/e$. Clearly, $c_k/e(e) \in \mathbb{N}^+$. The main steps in our FPTAS are listed as follows.

Step 1. Find positive real numbers $U_0$ and $L_0$ such that $L_0 < OPT(G) < U_0$ and $U_0 = 2L_0$, where $n = |V(G)|$. Note that the required $s^*t^*$-path in $G_k$ should satisfy the property $P$. Thus, in this step, we need to construct a path from $s^*$ to any other vertex $w_i$ with property $P$ and the minimal $d_0$-weight. Based on this, $U_0$ and $L_0$ can be obtained.

Step 2. Use a approximate test procedure to compute the lower or upper bounds of $OPT(G)$ more accurately.

Step 3. Begin with $U_0$ and $L_0$ given in Step 1, similar to the algorithm Rounding-AND-SEARCHING in [19], we can obtain $k$ disjoint st-paths $P_1, P_2, \ldots, P_k$ in $G$ with different colours such that $\sum_{i=1}^{k} c(P_i) \leq (1 + \epsilon)OPT(G)$ and $\sum_{i=1}^{k} d_i(P_i) \leq D$. That is $\sum_{i=1}^{k} c(P_i)$ is a $(1 + \epsilon)$-approximation solution.

4.2.1. Lower and upper bounds of $OPT(G)$

Let $c_1 < c_2 < \ldots < c^\theta$ be distinct $c_0$-weights of the edges in $G_k$. Then $G_{k1} \subseteq G_{k2} \subseteq \ldots \subseteq G_{k\theta}$ and $\eta \in |E_k|$. Recall that $w_1 = s^*$ and $w_{|V_k|} = t^*$. An $s^*t^*$-path $P$ with property $P$ is called a $D$-path if $d_0(P) \leq D$ holds.

First, for each $i \in [1, |V_k|]$ and $\beta \in [1, \eta]$, we consider the problem of finding an $s^*w_i$-path in $G_{ki\beta}$ with property $P$ and minimal $d_0$-weight. The $d_0$-weight of this path is denoted by $d_0^\beta(w_i)$. Similar to the concept $C_{w_i}$ in $G_k$, we can define $C_{w_i}^\beta$ in $G_{ki\beta}$. Given an $X = (x_1, x_2, \ldots, x_\lambda) \in C_{w_i}^\beta$, if there exists an $s^*w_i$-path $P$ in $G_{ki\beta}$ such that $col_\alpha(P) = x_\alpha$ when $col_\alpha(P) \neq 0$ for each $\alpha \in [1, k]$, let $d_0^\beta(w_i, X)$ be the minimal $d_0$-weight of this path; otherwise, let $d_0^\beta(w_i, X) = \infty$. Then, combining the property $P$ of $P$ with the shortest path algorithm in [1], we have that

$$d_0^\beta(w_i, X) = \min_{w_j \in N^\beta_{G_{ki\beta}}(w_i)} \{d_0^\beta(w_j, X) + d_0((w_j, w_i))\}.$$
So \( d^\beta(w_i) = \min_{X \in C'_w} \{ d^\beta_0(w_i, X) \} \).

Thus, combining this with the binary search, we can determine a \( \beta \in [1, \eta] \) such that \( d^\beta(t^*) \leq D \) and \( d^{\beta-1}(t^*) > D \). Then the time which is needed to find the \( \beta \) is \( \log \eta \) times the \( O((\binom{\eta}{k})E_k) \). Then, the following result holds.

**Lemma 4.2.** In \( O((\binom{\eta}{k})n^{k+1} \log n) \) time, we can find \( \beta \in [1, \eta] \) such that \( G_{kc^\beta} \) has a D-path and there is no D-path in \( G_{kc^\beta-1} \).

The lemma as follows can be used to compute the required lower and upper bounds of \( OPT(G) \).

**Lemma 4.3.** If there exists \( \beta \in [1, \eta] \) such that \( G_{kc^\beta} \) has a D-path and there is no D-path in \( G_{kc^\beta-1} \), then \( c^\beta < OPT(G) \leq (n - 2 + k)c^\beta < 2nc^\beta \).

**Proof.** Let \( P \) be a D-path in \( G_{kc^\beta} \) and \( P_j \) be the i-dimensional path of \( P \) for \( i = 1,2,\ldots,k \). Then, **Corollary 3.4** implies that \( P_1, P_2,\ldots,P_k \) are \( k \) internally disjoint st-path with different colours. So \( k \leq n - 2 \) and the number of edges of \( \bigcup_{j=1}^k P_j \) is smaller than or equal to \( n - 2 + k \) and larger than \( k \). Thus \( 2 \leq k < |E(P)| \leq n - 2 + k \leq 2n - 4 \). Combining this with that \( c_\beta(e) \leq c^\beta \) for each \( e \) in \( G_{kc^\beta} \), \( OPT(G) \leq c_\beta(P) \leq (n - 2 + k)c^\beta < 2nc^\beta \). As \( G_{kc^\beta-1} \) has no D-path, it can be seen that there exists an edge \( e \in E(P) \) such that \( c_\beta(e) = c^\beta \). Recall that \(|E(P)| \geq 2 \) and the \( c_\beta \)-weight of every edge in \( P \) is a positive number. Then \( c_\beta(P) > c^\beta \). By the arbitrariness of \( P \), it follows that \( OPT(G) > c^\beta \). \( \square \)

By Lemmas 4.2 and 4.3, the following corollary is immediate.

**Corollary 4.4.** In \( O((\binom{\eta}{k})n^{k+1} \log n) \) time, we can determine an upper bound \( U_0 \) and a lower bound \( L_0 \) of \( OPT(G) \) such that \( U_0 = 2nL_0 \).

4.2.2. The approximate scheme

We consider the approximate test procedure firstly.

For any given \( M, \delta \in \mathbb{R}^+ \), let \( r = M\delta/(2n) \). Then **Algorithm 1** can be used on \( G_{r} \) to compute \( M_{r}\{[2n/\delta]\} \) in \( O((\binom{\eta}{k})n^{k+2}\delta^{-1}) \) time. If \( M_{r}\{[2n/\delta]\} \leq D \) on \( G_{r} \), then there exists an \( s^* \)-path \( P \) with property \( P \) in \( G_{kc^\beta} \) such that \( \sum_{i=1}^k c_i(P_i) \leq [2n/\delta] \leq 2n/\delta \) and \( \sum_{i=1}^k d(P_i) \leq D \), where \( P_i \) is the i-dimensional path of \( P \). **Corollary 3.4** implies that \( P_1, P_2,\ldots,P_k \) are \( k \) vertex-disjoint st-paths in \( G_{r} \) with different colours and so \( OPT(G_{r}) \leq \sum_{i=1}^k c_i(P_i) \leq 2n/\delta \). If \( M_{r}\{[2n/\delta]\} > D \), then for any \( s^* \)-path \( P \) with property \( P \) in \( G_{r} \) such that \( \sum_{i=1}^k c_i(P_i) \leq 2n/\delta \), we have that \( \sum_{i=1}^k d(P_i) > D \). So \( OPT(G_{r}) > 2n/\delta \). Combining these with the TEST procedures in [3,19], we can observe that following lemma holds.

**Lemma 4.5.** Given \( M, \delta > 0 \), if \( M_{r}\{[2n/\delta]\} \leq D \) on \( G_{r} \), then \( OPT(G) \leq (1 + \delta)M \); otherwise, \( OPT(G) > M \).

Now, using the \( U_0 \) and \( L_0 \) as initial values, similar to the algorithm Rounding-And-Searching in [19], we can apply the **Lemma 4.5** repeatedly to obtain new lower and upper bounds \( L \) and \( U \) of \( OPT(G) \) such that \( U \leq 4L \). Then, set \( r = Le/(2n) \), where \( e > 0 \). Finally, **Algorithm 1** is used in \( G_{r} \) to compute \( OPT(G_{r}) \) and we can obtain the corresponding st-paths \( P_1, P_2,\ldots,P_k \) in \( G_{r} \). Then \( P_k \) also exists in \( G \) for \( i = 1,2,\ldots,k \) and \( \sum_{i=1}^k c(P_i) \) is a \((1 + \epsilon)\)-approximation solution. So, similar to the proofs in [3,19], we can summarise the main result in the following theorem.

**Theorem 4.6.** For any \( \epsilon > 0 \) and fixed \( \lambda \geq k > 1 \), \( k \) vertex-disjoint st-paths with different colours in a directed acyclic graph \( G \) such that total cost is at most \((1 + \epsilon)OPT(G) \) and total delay is at most \( D \) can be found in \( O((\binom{\eta}{k})n^{k+2}\epsilon^{-1}) \) time.

5. The restricted MinMax k-DPDC problem

Let \( OPT(G) \) be the total cost of an optimal solution of this problem on \( G \). Similar to Section 4, we first give an algorithm in subsection 5.1 for this problem on a DAG \( G \) with a partial \( \lambda \)-colouring and weight functions \( c: E(G) \rightarrow \mathbb{N}^+ \) and \( d: E(G) \rightarrow \mathbb{R}^+ \). Then, in subsection 5.2, we will use this algorithm and the technique named rounding-and-scaling to obtain the FPTAS of this problem when \( c(e) \in \mathbb{R}^+ \) for any \( e \in E(G) \).

5.1. Restricted MinMax k-DPDC problem when \( c(e) \in \mathbb{N}^+ \)

Following are details about the two major steps.

Step 1. For any positive integer \( C \) and any \( w_i \in V_k \), we find an \( s^* \)-path \( P \) with property \( P \) such that \( \max_{1 \leq j \leq k} c_j(P) \leq C \) and \( d_0(P) \) is minimised. Let \( M_{w_i}[C] = d_0(P) \).

Step 2. \( \min[C: M_{w_i}[C] \leq D] \) is the optimal solution \( OPT(G) \) for the restricted MinMax k-DPDC problem when the cost weight function is \( c : E(G) \rightarrow \mathbb{N}^+ \).
Procedure 2 The computation of $M_{w_i}[C]$.

1. $M_{w_i}[C] := M_{w_i}[C - 1];$
2. for any $C_1, C_2, \ldots, C_k \in [0, C]$ such that $\max_{1 \leq z \leq k} C_z = C$ and $N'_{w_i} \neq \emptyset$ do
3. $T_{w_i}[C_z : z \in [1, k]] := \infty;$
4. if $C_{w_i} \neq \emptyset$ do
5. for each $X \in C_w$, do
6. $T_{w_i}[C_z : z \in [1, k], X] := \infty;$
7. for each $w_j \in N'_{w_i}$ such that $X \in C_{w_i}$ do
8. if $T_{w_i}[C_z - c_z(w_j, w_i) : z \in [1, k], X] + d_0((w_j, w_i)) < T_{w_i}[C_z : z \in [1, k], X]$ do
9. $T_{w_i}[C_z : z \in [1, k], X] := T_{w_i}[C_z - c_z(w_j, w_i) : z \in [1, k], X] + d_0((w_j, w_i));$
10. end if
11. end for
12. if $T_{w_i}[C_z : z \in [1, k], X] < T_{w_i}[C_z : z \in [1, k]]$ do
13. $T_{w_i}[C_z : z \in [1, k]] := T_{w_i}[C_z : z \in [1, k], X];$
14. end if
15. end for
16. end if
17. if $T_{w_i}[C_z : z \in [1, k]] < M_{w_i}[C]$ do
18. $M_{w_i}[C] := T_{w_i}[C_z : z \in [1, k]];$
19. end if
20. end for

5.1.1. The computation of $M_{w_i}[C]$

Given any $w_i \in V_k$ and $k$ positive integers $C_1, C_2, \ldots, C_k$, choose $X = (x_1, x_2, \ldots, x_k) \in C_{w_i}$ (see the first paragraph of Section 4.1.1). If there is an $s^{w_i}$-paths $P$ in $G_k$ such that $col_a(P) = x_k$ when $col_a(P) \neq 0$ and $c_a(P) = c_A$ for each $\alpha \in [1, k]$, let $T_{w_i}[C_z : z \in [1, k], X]$ be the minimal $d_0$-weight of $P$; otherwise, let $T_{w_i}[C_z : z \in [1, k], X] = \infty$. Then, we can observe that if there exists a nonempty set $N'_{w_i} = \{w_j \in N^{-}_{G_k}(w_i) : j \geq 1 \text{ and } C_z - c_z((w_j, w_i)) \geq 0 \text{ for each } z \in [1, k]\}$, we have that

$$T_{w_i}[C_z : z \in [1, k], X] = \min_{w_j \in N'_{w_i}} \{T_{w_i}[C_z - c_z((w_j, w_i)) : z \in [1, k], X] + d_0((w_j, w_i))\}.$$

Now, if

$$T_{w_i}[C_z : z \in [1, k]] = \min_{X \in C_{w_i}} T_{w_i}[C_z : z \in [1, k], X],$$

then

$$M_{w_i}[C] = \min \{T_{w_i}[C_z : z \in [1, k]] : C_z \in [0, C] \text{ for every } z \in [1, k]\}$$

$$= \min \{M_{w_i}[C - 1], \max_{1 \leq z \leq k} \min T_{w_i}[C_z : z \in [1, k]] : \max C_z = C\}.$$ 

Therefore, we can use Procedure 2 to compute $T_{w_i}[C_z : z \in [1, k]]$ and $M_{w_i}[C]$.  

5.1.2. The computation of $OPT(G)$

In this subsection, we will compute $OPT(G)$ in the case that $c(e) \in \mathbb{N}^+$. Recall that $OPT(G) = \min \{C : M_{T^*}[C] \leq D\}$. By the definition of $M_{w_i}[C]$, we have that if there exists $C' \geq 1$ such that $M_{T^*}[C' - 1] > D$ and $M_{T^*}[C'] \leq D$, then $M_{T^*}[x] > D$ for each $x \in [1, C' - 1]$. So, $OPT(G) = C'$. Thus, it is need to compute $M_{T^*}[C]$ beginning with $C = 1$. If $M_{T^*}[C] > D$, then let $C := C + 1$ and continue to compute $M_{T^*}[C + 1]$; otherwise, $OPT(G) = C$. Therefore, the computation above can be described in the algorithm as follows.

In the following, we compute the time complexity of Algorithm 2. Lemma 3.1 implies that it needs $O(n^{k+1})$ time to construct $G_k$ in Line 1. The topological order of $V_k$ in Line 2 can be obtained in $O(|V_k| + |E_k|) = O(n^{k+1})$ time. Lines 3–4 totally take $O(k^2)$ time. In addition, $O((\binom{k}{2})|V_k|) = O((\binom{k}{2})n^{k+1})$ is needed in Lines 5–10. Note that the number of iterations of the for loop in lines 12–14 is

$$|(\{C_1, C_2, \ldots, C_k\} : C_z \in [0, C] \text{ for each } z \in [1, k] \text{ and } \max_{1 \leq z \leq k} C_z = C)|$$

$$= O(k^2(C + 1)^k - 1).$$
Algorithm 2 Restricted MinMax k-DPDC problem when $c(e) \in \mathbb{N}^+$. 

Input: A DAG $G$ with a partial $\lambda$-colouring and weight functions $c: E(G) \to \mathbb{N}^+$ and $d: E(G) \to \mathbb{R}^+$, two vertices $s, t$ with colour $0$ and upper bound $D$. 

Output: the optimal solution

1. construct $G_0 = (V_0, E_0)$ with weight functions $c_0, d_0$ and $c_0(\alpha) = [1, k]$;
2. find the topological ordering $w_1, w_2, \ldots, w_{|V|}$ of the vertices in $V_G$;
3. let $C_{x} = \{x_1, x_2, \ldots, x_6\}$ such that $x_4 = \alpha(x) \leq \lambda$ for any $x \neq b$ and there exists $w \in N_{x}^C(s^*)$ such that $x_4 = c_0(w)$ when $c_0(w) \neq 0$ for each $\alpha \in [1, k]$;
4. $T_{x} = 0; z \in [1, k], X := 0$ for each $X \in C_{x}$;
5. for $i = 2$ to $|V| - 1$ do;
6. $C := \bigcup_{|V|, \alpha} C_{x}$;
7. $C_{x} := \{x_1, x_2, \ldots, x_6\} \in C : x_4 = c_0(w)$ when $c_0(w) \neq 0$ for each $\alpha \in [1, k]$;
8. $T_{x}[C] := 0: z \in [1, k], X := 0$ for each $X \in C_{x}$;
9. $M_{x}[0] := \infty$;
10. end for;
11. for each $C_{1}, C_{2}, \ldots, C_{k} \in [0, C]$ such that $\max_{1 \leq i \leq k} C_{i} = C$
12. $T_{x}[C] := z \in [1, k], X := 0$ for each $X \in C_{x}$;
13. end for;
14. for $i = 2$ to $|V| - 1$ do;
15. use Procedure 2 to compute $M_{x}[C]$;
16. if $w_i = \ell^*$ and $M_{x}[C] \leq D$ do
17. return OPT($G$) = $C$ and its corresponding st-paths, exit;
18. end if;
19. end for;
20. C := C + 1 and return to Step 12.

As the time complexity of Line 13 is $O\left(\binom{C}{2}\right)$, we have that this for loop needs $O\left(k(C + 1)^{k-1}\binom{C}{2}\right) = O\left(k(C + 1)^{k-1}n\right)$ time. We can observe that for any $w_i$, Line 16 takes $O\left(\binom{C}{2}(C + 1)^{k-1}\big|N_{w_i}^C(w)\right)$ time to compute $M_{w_i}[C]$ by Procedure 2. Then the total time for Lines 15–20 is $\sum_{i=2,|V|}-\binom{C}{2}(C + 1)^{k-1}\big|N_{w_i}^C(w)\right) = O\left(\binom{C}{2}(C + 1)^{k-1}n\right)$. Therefore, Lines 21–22 can be done within $O\left(\sum_{C=1}^{\text{OPT}(G)}\binom{C}{2}(C + 1)^{k-1}\big|N_{w_i}^C(w)\right) = O\left(\binom{C}{2}(C + 1)^{k-1}n^k\right)$. So, following result holds obviously.

**Theorem 5.1.** For any fixed $\lambda \geq k > 1$, when $c(e) \in \mathbb{N}^+$, the Restricted MinMax k-DPDC problem on a directed acyclic graph can be solved in $O\left(\binom{C}{2}(C + 1)^{k-1}n^{k+2}\right)$ time.

5.2. The FPTAS

Now we consider the restricted MinMax k-DPDC problem on $G$ when $c(e) \in \mathbb{R}^+$. 

5.2.1. Lower and upper bounds of OPT($G$)

Recall that $c^1 < c^2 < \ldots < c^\ell$ are distinct $c_0$-weights of the edges in $E_k$. Then we have following result.

**Lemma 5.2.** If there exists $\beta \in [1, \eta]$ such that $G_{k\beta}$ has a D-path and there is no D-path in $G_{k\beta-1}$, then $c^\beta \leq \text{OPT}(G) < nc^\beta$.

**Proof.** Let $P$ be a D-path in $G_{k\beta}$, $P_i$ be the i-dimensional path of $P$ for $i = 1, 2, \ldots, k$ and let $\max_{1 \leq i \leq k} c(P_i) = \text{OPT}(G)$. Without loss of generality, we may assume that $c(P_1) = c_1(P) = \text{OPT}(G)$. Note that $1 \leq |E(P_1)| < n$ and $c_1(e) \leq c_0(e) \leq c^\beta$ for each $e$ of $G_{k\beta}$. Then $\text{OPT}(G) = c_1(P) < nc^\beta$. As $G_{k\beta-1}$ has no D-path, it can be seen that there exists an edge $e \in E(P)$ such that $c_0(e) = c^\beta$. Assume that the corresponding edge of $e$ is in $P_j$, where $1 \leq j \leq k$. Then $c_j(P) = \max_{1 \leq i \leq k} c_j(P_i) \geq c_j(e) = c_0(e) = c^\beta$ and so $\text{OPT}(G) \geq c^\beta$. □

Combining this with Lemma 4.2, the following corollary holds.

**Corollary 5.3.** In $O\left(\binom{n}{2}n^{k+1}n^{k+1}\right)$ time, we can determine an upper bound $U_0$ and a lower bound $L_0$ of $\text{OPT}(G)$ such that $U_0 = nL_0$.

5.2.2. The approximate scheme

Given $M, \delta > 0$, then the value of $M_{r}\left[\left[\eta/\delta\right]\right]$ on $G_{r}$ can be computed by Algorithm 2 in $O\left(\binom{n}{2}n^{2k^2+2}\left(1/\delta^k\right)^k\right)$ time. If $M_{r}\left[\left[\eta/\delta\right]\right] \leq D$ on $G_r$, then there are $k$ disjoint st-path $P_1, P_2, \ldots, P_k$ with different colours in $G_{r}$ such that $\text{OPT}(G_{r}) \leq \max_{1 \leq i \leq k} c_{r}(P_i) \leq \left[\eta/\delta\right] \leq n/\delta$ and $\sum_{i=1}^{k} d(P_i) \leq D$. If $M_{r}\left[\left[\eta/\delta\right]\right] > D$, then for any $k$ disjoint st-path $P_1, P_2, \ldots, P_k$ with
different colours in $G_{fr}$ such that $\max_{1 \leq i \leq k} c_{fr}(P_i) \leq \lceil n/\delta \rceil$, we have that $\sum_{1 \leq i \leq k} d(P_i) > D$. So $OPT(G_{fr}) > n/\delta$. Therefore, from what has been discussed above and the TEST procedures in [3,19], following result is immediate.

**Lemma 5.4.** Given $M, \delta > 0$, let $r = M\delta/n$. If $M_{fr}([[n/\delta]]) \leq D$ on $G_{fr}$, then $OPT(G) \leq (1 + \delta)M$; otherwise, $OPT(G) > M$.

Now, use Corollary 5.3 and Lemma 5.4 in the algorithm ROUNDOING-AND-SEARCHING in [19], the main result of this section is as follows.

**Theorem 5.5.** For any $\epsilon > 0$ and fixed $\lambda \geq k > 1$, $k$ vertex-disjoint $st$-paths $P_1, P_2, \ldots, P_k$ with different colours in a directed acyclic graph $G$ such that $\max_{1 \leq i \leq k} c(P_i) \leq (1 + \epsilon)OPT(G)$ and $\sum_{i=1}^{k} d(P_i) \leq D$ can be found in $O((\lambda^k)^n k^{2k+2} (1/\epsilon)^k)$ time.

### 6. Conclusions

In this paper, motivated by the expectation that the data packets can be transmitted effectively from source node to sink node when some channel problems or node faults occur, we introduced and investigated the restricted $k$-DPDC problem on multi-channel wireless networks and considered two important objectives MinSum and MinMax when the network topology is an acyclic directed graph. For these two objectives, we supposed the cost of each edge is a positive integer firstly and proposed algorithms to obtained optimal solutions. Then, we used these algorithms and the technique named rounding-and-scaling to present FPTAS for the corresponding objectives when the cost weight of each edge is a positive real number. Because of the widespread use of multi-channel wireless networks and the importance of the fault tolerance and effectiveness, we aim at extending our efforts towards the design and optimisation of multiple node- and channel-disjoint multicast tree problems in future studies.

### Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments that improve the quality of this paper.

### References