Theoretical Computer Science 695 (2017) 74–82

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

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

Shurong Zhang [a,*], Lin Chen [b], Weihua Yang [a]

[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

ARTICLE INFO

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

ABSTRACT

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.

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 $k \le \lambda$ 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 P_1 and P_2 are channel-disjoint if there exist distinct c_1 and c_2 with $1 \le c_1, c_2 \le \lambda$ such that P_i $(i = 1, 2)$ is a channel c_i 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/

- **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 λ 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 λ-vertex colouring* of G is a mapping $col : V(G) \to \{0, 1, \ldots, \lambda\}$ such that there exist at least two vertices of the colour 0. If $col(v) = 0$ for every $v \in V(G)$, the partial λ-vertex colouring of G is called an *empty λ-colouring*. Let $P = \langle v_1, v_2, \ldots, v_l\rangle$ be a path in G. If $col(v_i) = 0$ for $i = 0, 1, \ldots, l$, set $col(P) = 0$; if the colour of some vertex in P is not 0 and there exists a colour $c \in [1, \lambda]$ such that $col(v_i) \in \{0, c\}$ for $i = 0, 1, \ldots, l$, set $col(P) = c$.

Definition 2.1 *(k-DPDC problem).* Given a graph $G = (V(G), E(G))$ with a partial λ-vertex colouring such that both the source vertex s and target vertex t have colour 0. Let $k \le \lambda$ 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, $col(P_i)$ exists and for any two different paths P_i and P_j, we must have $col(P_i) \neq col(P_j)$ or $col(P_i) = col(P_j) = 0$.

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 λ-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 λ-vertex colouring, two distinct vertices $s, t \in V(G)$ with colour 0, a positive integer D, a cost function $c : E(G) \to \mathbb{R}^+$ and a delay function $d : E(G) \to \mathbb{R}^+$, 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 $\sum_{i=1}^{k} d(P_i) \le D$, 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 a graph with an empty λ-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 ($k \ge 2$) on a DAG in Section 4 of this paper.

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

When the partial λ-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 strong 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

$V(G)$ and the directed edge set $E(G)$. For any two vertices $u, v \in V(G)$, the directed edge $(u, v) \in E(G)$ if and only if a data packet can be transmitted from u to v in the network. Let $|V(G)| = n$ and $|E(G)| = m$. Each edge $e \in E(G)$ has an associated cost $c(e) \in \mathbb{R}^+$ and a delay $d(e) \in \mathbb{R}^+$. $N_G^-(v) = \{u \in V(G - v) : (u, v) \in E(G)\}$ is called the *in-neighbourhood* of v. Similarly, $N_G^+(v) = \{u \in V(G - v) : (v, u) \in E(G)\}$ is called the *out-neighbourhood* of v. For graph-theoretical terminology and notation not defined here we follow [1].

In this paper, we suppose that G is a DAG. Then, let $u_1, u_2, \dots, u_{|V(G)|}$ be a topological ordering of the vertices in $V(G)$, i. e. $(u_i, u_j) \in E(G)$ only if $i < j$ [2]. Choose two vertices $s, t \in V(G)$. Now we introduce an auxiliary directed graph $G_k = (V_k, E_k)$ [10,12] of G as follows.

$$V_k = \{\langle v_1, v_2, \dots, v_k\rangle : v_i \in V(G) \text{ and } v_i \neq v_j \text{ or } v_i = v_j \in \{s, t\} \text{ for any } i \neq j\},$$
$$E_k = \{(\langle v_1, \dots v_{d-1}, v_d, v_{d+1} \dots, v_k\rangle, \langle v_1, \dots v_{d-1}, v'_d, v_{d+1} \dots, v_k\rangle) : (v_d, v'_d) \in E(G) \text{ and}$$
$$f(d) = \min_{1\le i\le k} f(i) \text{ where } f(x) = y \text{ if } v_x = u_y \text{ for } 1 \le x \le k \text{ and } 1 \le y \le |V(G)|\}.$$

Then $s^* = \langle s, s, \dots, s\rangle \in V_k$ and $t^* = \langle t, t, \dots, t\rangle \in V_k$.

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 $e = (\langle v_1, \dots v_{d-1}, v_d, v_{d+1} \dots, v_k\rangle, \langle v_1, \dots v_{d-1}, v'_d, v_{d+1} \dots, v_k\rangle) \in E_k$ 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 $c_0(e), c_1(e), c_2(e), \dots, c_k(e)$, 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 $P_1, P_2, \dots, P_k$ 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 $i = 1, 2, \dots, k$ 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 $C = (\langle u_1, v_1\rangle, \langle u_2, v_1\rangle, \dots, \langle u_i, v_j\rangle, \langle u_i, v_{j+1}\rangle, \dots, \langle u_1, v_1\rangle)$ 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 $C_1 = (u_1, u_2, \dots, u_i, \dots, u_1)$ and $C_2 = (v_1, \dots, v_j, v_{j+1}, \dots, v_1)$. Obviously, $|N_{C_i}^-(w)| = |N_{C_i}^+(w)|$ 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. $\square$

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

For any vertex $v = \langle v_1, v_2, \dots, v_k\rangle \in V_k$ 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* $\mathcal{P}$ 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 $P_1, P_2, \dots, P_k$ with different colours in G if and only if there is an s^*t^*-path P with property $\mathcal{P}$ in G_k and P_i $(i = 1, 2, \dots, k)$ 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.

Procedure 1 The computation of $M_{w_i}[C]$.

```
$M_{w_i}[C] := M_{w_i}[C-1]$, $T_{w_i}[C] := \infty$;
if $\mathcal{C}_{w_i} \neq \emptyset$ and $N_{w_i} \neq \emptyset$, do
  for each $X \in \mathcal{C}_{w_i}$, do
    $T_{w_i}[C, X] := \infty$;
    for each $w_j \in N_{w_i}$ such that $X \in \mathcal{C}_{w_j}$, do
      if $T_{w_j}[C - c_0((w_j, w_i)), X] + d_0((w_j, w_i)) < T_{w_i}[C, X]$, do
        $T_{w_i}[C, X] := T_{w_j}[C - c_0((w_j, w_i)), X] + d_0((w_j, w_i))$;
      end if
    end for
    if $T_{w_i}[C, X] < T_{w_i}[C]$, do
      $T_{w_i}[C] := T_{w_i}[C, X]$;
    end if
  end for
end if
```

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 $OPT(G)$ of the restricted MinSum k-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 $\min\{C : M_{t^*}[C] \leq D\}$ is the optimal solution $OPT(G)$ 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 $\mathcal{C}_{w_i}$ be a subset of $U = \{\langle x_1, x_2, \ldots, x_k\rangle : 1 \leq x_a \neq x_b \leq \lambda$ for any $a \neq b\}$ such that $\langle x_1, x_2, \ldots, x_k\rangle \in \mathcal{C}_{w_i}$ if and only if there exists an s^*w_i-path P in G_k such that $col_\alpha(P) = x_\alpha$ when $col_\alpha(P) \neq 0$ for each $\alpha \in [1, k]$.

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

$$\mathcal{C}_{w_i} = \{\langle x_1, x_2, \ldots, x_k\rangle \in \mathcal{C} : col_\alpha(w_i) = x_\alpha \text{ when } col_\alpha(w_i) \neq 0 \text{ for each } \alpha \in [1, k]\}.$$

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 $\mathcal{C}_{w_i}$. Suppose that $\mathcal{C}_{w_i} \neq \emptyset$ and $i \geq 2$. Choose $X = \langle x_1, x_2, \ldots, x_k\rangle \in \mathcal{C}_{w_i}$. If there exists an s^*w_i-path P with property $\mathcal{P}$ such that $c_0(P) = C$ and $col_\alpha(P) = x_\alpha$ when $col_\alpha(P) \neq 0$ 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 $T_{w_1}[0, X] = 0$.

Now, if we may choose a nonempty vertex subset

$$N_{w_i} = \{w_j \in N^-_{G_k}(w_i) : j \geq 1,\ C - c_0((w_j, w_i)) \geq 0\},$$

we can observe that 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 $T_{w_i}[C] = \min_{X \in \mathcal{C}_{w_i}}\{T_{w_i}[C, X]\}$, then

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

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

4.1.2. The computation of $OPT(G)$

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 $OPT(G) = \min\{C : M_{t^*}[C] \leq D\}$. Therefore, begin with $C = 1$. If $M_{t^*}[C] \leq D$, then $OPT(G) = C$ 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 $O(|V_k| + |E_k|)$ time to obtain a topological order of V_k. Clearly, Lines 3–4 take $O(\binom{\lambda}{k})$ time. Since times $O(N^-_{G_k}(w_i))$ and $O(\binom{\lambda}{k})$ 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 λ-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 $G_k = (V_k, E_k)$ with weight functions c_0, d_0 and col_α 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 $\mathcal{C}_{s^*} = \{\langle x_1, x_2, \ldots, x_k\rangle : 1 \le x_a \neq x_b \le \lambda$ for any $a \neq b$ and there exists $w \in N^+_{G_k}(s^*)$ such that $x_\alpha = col_\alpha(w)$ when $col_\alpha(w) \neq 0$ for each $\alpha \in [1, k]\}$
4. $T_{s^*}[0, X] := 0$ for each $X \in \mathcal{C}_{s^*}$;
5. **for** $i = 2$ to $|V_k|$ **do**
6. $\quad \mathcal{C} := \bigcup_{w_j \in N^-_{G_k}(w_i)} \mathcal{C}_{w_j}$;
7. $\quad \mathcal{C}_{w_i} := \{\langle x_1, x_2, \ldots, x_k\rangle \in \mathcal{C} : x_\alpha = col_\alpha(w_i)$ when $col_\alpha(w_i) \neq 0$ for each $\alpha \in [1, k]\}$;
8. $\quad T_{w_i}[0, X] := \infty$ for each $X \in \mathcal{C}_{w_i}$;
9. $\quad M_{w_i}[0] := \infty$;
10. **end for**
11. $C := 1$;
12. **for** $i = 2$ to $|V_k|$ **do**
13. $\quad$ use Procedure 1 to compute $M_{w_i}[C]$;
14. $\quad$ **if** $w_i = t^*$ and $M_{t^*}[C] \le D$ **do**
15. $\quad\quad$ return $OPT(G) = C$ and its corresponding paths, exit;
16. $\quad$ **end if**
17. **end for**
18. $C := C + 1$ and return to step 12.

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

Theorem 4.1. *For any fixed $\lambda \ge k > 1$, when $c(e) \in \mathbb{N}^+$, the Restricted MinSum k-DPDC problem on a directed acyclic graph can be solved in $O(\binom{\lambda}{k}n^{k+1}OPT(G))$ 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_{kx} be a graph obtained from G_k by deleting the edge set $\{e \in E_k : c_0(e) > x\}$. If we replace weight $c(e)$ of any $e \in E(G)$ with $c_{/x}(e) = \lfloor c(e)/x \rfloor$, then we obtain a new weighted graph, denoted by $G_{/x}$. Clearly, $c_{/x}(e) \in \mathbb{N}^+$. 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 $U_0 = 2nL_0$, where $n = |V(G)|$. Note that the required s^*t^*-path in G_k should satisfy the property $\mathcal{P}$. Thus, in this step, we need to construct a path from s^* to any other vertex w_i with property $\mathcal{P}$ 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 $\sum_{i=1}^k c(P_i) \le (1+\epsilon)OPT(G)$ and $\sum_{i=1}^k d(P_i) \le D$. 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 $c^1 < c^2 < \ldots < c^\eta$ be distinct c_0-weights of the edges in G_k. Then $G_{kc^1} \subset G_{kc^2} \subset \ldots \subset G_{kc^\eta}$ and $\eta \le |E_k|$. Recall that $w_1 = s^*$ and $w_{|V_k|} = t^*$. An s^*t^*-path P with property $\mathcal{P}$ is called a D-path if $d_0(P) \le D$ holds.

First, for each $i \in [1, |V_k|]$ and $\beta \in [1, \eta]$, we consider about the problem of finding an s^*w_i-path in G_{kc^β} with property $\mathcal{P}$ and minimal d_0-weight. The d_0-weight of this path is denoted by $d_0^\beta(w_i)$. Similar to the concept $\mathcal{C}_{w_i}$ in G_k, we can define $\mathcal{C}^\beta_{w_i}$ in G_{kc^β}. Given an $X = \langle x_1, x_2, \ldots, x_k\rangle \in \mathcal{C}^\beta_{w_i}$, if there exists an s^*w_i-path P in G_{kc^β} 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 $\mathcal{P}$ of P with the shortest path algorithm in [1], we have that

$$d_0^\beta(w_i, X) = \min_{w_j \in N^-_{G_{kc^\beta}}(w_i)} \{d_0^\beta(w_j, X) + d_0((w_j, w_i))\}.$$

So $d_0^{\beta}(w_i) = \min_{X \in C_{w_i}^{\beta}} \{d_0^{\beta}(w_i, X)\}$.

Thus, combining this with the binary search, we can determine a $\beta \in [1, \eta]$ such that $d_0^{\beta}(t^*) \le D$ and $d_0^{\beta-1}(t^*) > D$. Then the time which is needed to find the β is $log\eta$ times the $O(\binom{\lambda}{k}|E_k|)$. Then, the following result holds.

Lemma 4.2. *In $O(\binom{\lambda}{k}n^{k+1}logn)$ 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 $c^{\beta} < OPT(G) \le (n-2+k)c^{\beta} < 2nc^{\beta}$.*

Proof. Let P be a D-path in $G_{kc^{\beta}}$ and P_i be the i-dimensional path of P for $i = 1, 2, \dots, k$. Then, Corollary 3.4 implies that $P_1, P_2, \dots, P_k$ are k internally disjoint st-path with different colours. So $k \le n-2$ and the number of edges of $\bigcup_{i=1}^{k} P_i$ is smaller than or equal to $n-2+k$ and lager than k. Thus $2 \le k < |E(P)| \le n-2+k \le 2n-4$. Combining this with that $c_0(e) \le c^{\beta}$ for each e in $G_{kc^{\beta}}$, $OPT(G) \le c_0(P) \le (n-2+k)c^{\beta} < 2nc^{\beta}$. As $G_{kc^{\beta-1}}$ has no D-path, it can be seen that there exists an edge $e \in E(P)$ such that $c_0(e) = c^{\beta}$. Recall that $|E(P)| \ge 2$ and the c_0-weight of every edge in P is a positive number. Then $c_0(P) > c^{\beta}$. By the arbitrariness of P, it follows that $OPT(G) > c^{\beta}$. □

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

Corollary 4.4. *In $O(\binom{\lambda}{k}n^{k+1}logn)$ 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 $M_{t^*}[\lfloor 2n/\delta \rfloor]$ in $O(\binom{\lambda}{k}n^{k+2}\delta^{-1})$ time. If $M_{t^*}[\lfloor 2n/\delta \rfloor] \le D$ on $G_{/r}$, then there exists an s^*t^*-path P with property $\mathcal{P}$ in $G_{/rk}$ such that $\sum_{i=1}^{k} c_{/r}(P_i) \le \lfloor 2n/\delta \rfloor \le 2n/\delta$ and $\sum_{i=1}^{k} d(P_i) \le D$, where P_i is the i-dimensional path of P. Corollary 3.4 implies that $P_1, P_2, \dots, P_k$ are k vertex-disjoint st-paths in $G_{/r}$ with different colours and so $OPT(G_{/r}) \le \sum_{i=1}^{k} c_{/r}(P_i) \le 2n/\delta$. If $M_{t^*}[\lfloor 2n/\delta \rfloor] > D$, then for any s^*t^*-path P with property $\mathcal{P}$ in $G_{/r}$ such that $\sum_{i=1}^{k} c_{/r}(P_i) \le 2n/\delta$, 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 $M_{t^*}[\lfloor 2n/\delta \rfloor] \le D$ on $G_{/r}$, then $OPT(G) \le (1+\delta)M$; 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 $U \le 4L$. Then, set $r = L\epsilon/(2n)$, where $\epsilon > 0$. Finally, Algorithm 1 is used in $G_{/r}$ to compute $OPT(G_{/r})$ and we can obtain the corresponding st-paths $P_1, P_2, \dots, P_k$ in $G_{/r}$. Then P_i also exists in G for $i = 1, 2, \dots, k$ 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 $\lambda \ge k > 1$, 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 $O(\binom{\lambda}{k}n^{k+2}\epsilon^{-1})$ time.*

5. The restricted MinMax *k*-DPDC problem

Let $OPT(G)$ be the total cost of a 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 λ-colouring and weight functions $c: E(G) \to \mathbb{N}^+$ and $d: E(G) \to \mathbb{R}^+$. 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^*t^*-path P with property $\mathcal{P}$ such that $\max_{1 \le j \le k} c_j(P) \le C$ and $d_0(P)$ is minimised. Let $M_{w_i}[C] = d_0(P)$.

Step 2. $\min\{C : M_{t^*}[C] \le D\}$ is the optimal solution $OPT(G)$ for the restricted MinMax k-DPDC problem when the cost weight function is $c: E(G) \to \mathbb{N}^+$.

Procedure 2 The computation of $M_{w_i}[C]$.

$M_{w_i}[C] := M_{w_i}[C-1]$;
for any $C_1, C_2, \ldots, C_k \in [0, C]$ such that $\max_{1 \le z \le k} C_z = C$ and $N'_{w_i} \neq \emptyset$ **do**
  $T_{w_i}[C_z : z \in [1,k]] := \infty$;
  if $\mathcal{C}_{w_i} \neq \emptyset$ **do**
    for each $X \in \mathcal{C}_{w_i}$ **do**
      $T_{w_i}[C_z : z \in [1,k], X] := \infty$;
      for each $w_j \in N'_{w_i}$ such that $X \in \mathcal{C}_{w_j}$ **do**
        if $T_{w_j}[C_z - c_z((w_j, w_i)) : z \in [1,k], X] + d_0((w_j, w_i)) < T_{w_i}[C_z : z \in [1,k], X]$ **do**
          $T_{w_i}[C_z : z \in [1,k], X] := T_{w_j}[C_z - c_z((w_j, w_i)) : z \in [1,k], X] + d_0((w_j, w_i))$;
        end if
      end for
      if $T_{w_i}[C_z : z \in [1,k], X] < T_{w_i}[C_z : z \in [1,k]]$ **do**
        $T_{w_i}[C_z : z \in [1,k]] := T_{w_i}[C_z : z \in [1,k], X]$;
      end if
    end for
  end if
  if $T_{w_i}[C_z : z \in [1,k]] < M_{w_i}[C]$ **do**
    $M_{w_i}[C] := T_{w_i}[C_z : z \in [1,k]]$;
  end if
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 $X = \langle x_1, x_2, \ldots, x_k \rangle \in \mathcal{C}_{w_i}$ (see the first paragraph of Section 4.1.1). If there is an $s^* w_i$-paths P in G_k such that $col_\alpha(P) = x_\alpha$ when $col_\alpha(P) \neq 0$ and $c_\alpha(P) = C_\alpha$ 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

$$N'_{w_i} = \{w_j \in N^-_{G_k}(w_i) : j \ge 1 \text{ and } C_z - c_z((w_j, w_i)) \ge 0 \text{ for each } z \in [1,k]\},$$

we have that

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

Now, if

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

then

$$\begin{aligned} M_{w_i}[C] &= \min\{T_{w_i}[C_z : z \in [1,k]] : C_z \in [0, C] \text{ for every } z \in [1,k]\} \\ &= \min\{M_{w_i}[C-1], \min\{T_{w_i}[C_z : z \in [1,k]] : \max_{1 \le z \le k} C_z = C\}\}. \end{aligned}$$

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 $OPT(G) = \min\{C : M_{t^*}[C] \le D\}$. By the definition of $M_{w_i}[C]$, we have that if there exists $C' \ge 1$ such that $M_{t^*}[C'-1] > D$ and $M_{t^*}[C'] \le D$, 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 $O(\binom{\lambda}{k})$ time. In addition, $O(\binom{\lambda}{k}|V_k|) = O(\binom{\lambda}{k} n^{k+1})$ is needed in Lines 5–10. Note that the number of iterations of the for loop in lines 12–14 is

$$\begin{aligned} & |\{\{C_1, C_2, \ldots, C_k\} : C_z \in [0, C] \text{ for each } z \in [1,k] \text{ and } \max_{1 \le z \le k} C_z = C\}| \\ & \quad = O(k(C+1)^{k-1}). \end{aligned}$$

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

Input: A DAG G with a partial λ-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 $G_k = (V_k, E_k)$ with weight functions c_0, d_0 and col_α ($\alpha \in [1, k]$);
2. find the topological ordering $w_1, w_2, \ldots, w_{|V_k|}$ of the vertices in V_k;
3. let $\mathcal{C}_{s^*} = \{\langle x_1, x_2, \ldots, x_k\rangle : 1 \le x_a \ne x_b \le \lambda$ for any $a \ne b$ and there exists $w \in N^+_{G_k}(s^*)$ such that $x_\alpha = col_\alpha(w)$ when $col_\alpha(w) \ne 0$ for each $\alpha \in [1, k]\}$
4. $T_{s^*}[C_z = 0 : z \in [1, k], X] := 0$ for each $X \in \mathcal{C}_{s^*}$;
5. **for** $i = 2$ to $|V_k|$ **do**;
6. $\quad \mathcal{C} := \bigcup_{w_j \in N^-_{G_k}(w_i)} \mathcal{C}_{w_j}$;
7. $\quad \mathcal{C}_{w_i} := \{\langle x_1, x_2, \ldots, x_k\rangle \in \mathcal{C} : x_\alpha = col_\alpha(w_i)$ when $col_\alpha(w_i) \ne 0$ for each $\alpha \in [1, k]\}$;
8. $\quad T_{w_i}[C_z = 0 : z \in [1, k], X] := \infty$ for each $X \in \mathcal{C}_{w_i}$;
9. $\quad M_{w_i}[0] := \infty$;
10. **end for**
11. $C := 1$;
12. **for** each $C_1, C_2, \ldots, C_k \in [0, C]$ such that $\max_{1 \le x \le k} C_k = C$ **do**
13. $\quad T_{s^*}[C_z : z \in [1, k], X] := \infty$ for each $X \in \mathcal{C}_{s^*}$;
14. **end for**
15. **for** $i = 2$ to $|V_k|$ **do**
16. $\quad$ use Procedure 2 to compute $M_{w_i}[C]$;
17. $\quad$ **if** $w_i = t^*$ and $M_{w_i}[C] \le D$ **do**
18. $\quad\quad$ return $OPT(G) = C$ and its corresponding st-paths, exit;
19. $\quad$ **end if**
20. **end for**
21. $C := C + 1$ and return to Step 12.

As the time complexity of Line 13 is $O(\binom{\lambda}{k})$, we have that this for loop needs $O(k(C+1)^{k-1}\binom{\lambda}{k}) = O(\binom{\lambda}{k}(C+1)^{k-1}n)$ time. We can observe that for any w_i, Line 16 takes $O(\binom{\lambda}{k}k(C+1)^{k-1}|N^-_{G_k}(w_i)|)$ time to compute $M_{w_i}[C]$ by Procedure 2. Then the total time for Lines 15–20 is $\sum_{i \in [2,|V_k|]} O(\binom{\lambda}{k}k(C+1)^{k-1}|N^-_{G_k}(w_i)|) = O(\binom{\lambda}{k}k(C+1)^{k-1}|E_k|) = O(\binom{\lambda}{k}(C+1)^{k-1}n^{k+2})$. Therefore, Lines 12–21 can be done within $O(\sum_{C=1}^{OPT(G)}(\binom{\lambda}{k}(C+1)^{k-1}n^{k+2})) = O(\binom{\lambda}{k}n^{k+2}(OPT(G)+1)^k)$. So, following result holds obviously.

Theorem 5.1. *For any fixed $\lambda \ge k > 1$, when $c(e) \in \mathbb{N}^+$, the Restricted MinMax k-DPDC problem on a directed acyclic graph can be solved in $O(\binom{\lambda}{k}n^{k+2}(OPT(G)+1)^k)$ 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 $c^1 < c^2 < \ldots < c^\eta$ 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_{kc^β} has a D-path and there is no D-path in $G_{kc^{\beta-1}}$, then $c^\beta \le OPT(G) < nc^\beta$.*

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

Combining this with Lemma 4.2, the following corollary holds.

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

5.2.2. The approximate scheme

Given $M, \delta > 0$, then the value of $M_{t^*}[\lfloor n/\delta \rfloor]$ on $G_{/r}$ can be computed by Algorithm 2 in $O(\binom{\lambda}{k}n^{2k+2}(1/\delta)^k)$ time. If $M_{t^*}[\lfloor n/\delta \rfloor] \le D$ 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 $OPT(G_{/r}) \le \max_{1 \le i \le k} c_{/r}(P_i) \le \lfloor n/\delta \rfloor \le n/\delta$ and $\sum_{i=1}^k d(P_i) \le D$. If $M_{t^*}[\lfloor n/\delta \rfloor] > D$, then for any k disjoint st-path $P_1, P_2, \ldots, P_k$ with different colours in $G_{/r}$ such that $\max_{1\le i\le k} c_{/r}(P_i) \le \lfloor n/\delta \rfloor$, we have that $\sum_{1\le i\le k} d(P_i) > D$. So $OPT(G_{/r}) > n/\delta$. 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 $M_{t^*}[\lfloor n/\delta \rfloor] \le D$ on $G_{/r}$, then $OPT(G) \le (1+\delta)M$; otherwise, $OPT(G) > M$.*

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

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

[1] J.A. Bondy, U.S.R. Murty, Graph Theory, Springer, New York, 2007.
[2] Jørgen Bang-Jensen, Gregory Gutin, Digraphs Theory, Algorithms and Applications, Springer-Verlag, London, 2007.
[3] Funda Ergun, Rakesh Sinha, Lisa Zhang, An improved FPTAS for restricted shortest path, Inform. Process. Lett. 83 (2002) 287–291.
[4] Rudolf Fleischer, Qi Ge, Jian Li, Hong Zhu, Efficient algorithms for k-disjoint paths problems on DAGs, in: Proceedings of the 3rd International Conference on Algorithmic Aspects in Information and Management, 2007, pp. 134–143.
[5] Michael R. Garey, David S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, San Francisco, 1979.
[6] Refael Hassin, Approximation schemes for the restricted shortest path problem, Math. Oper. Res. 17 (1) (1992) 36–42.
[7] Oscar H. Ibarra, Chul E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, J. Assoc. Comput. Mach. 22 (4) (1975) 463–468.
[8] A. Itai, Y. Perl, Y. Shiloach, The complexity of finding maximum disjoint paths with length constraints, Networks 12 (3) (1982) 277–286.
[9] Eugene L. Lawler, Fast approximation algorithms for knapsack problems, Math. Oper. Res. 4 (4) (1979) 339–356.
[10] Chung-Lun Li, S. Thomas McCormick, David Simchi-Levi, The complexity of finding two disjoint paths with Min–Max objective function, Discrete Appl. Math. 26 (1) (1990) 105–115.
[11] Dean H. Lorenz, Danny Raz, A simple efficient approximation scheme for the restricted shortest path problem, Oper. Res. Lett. 28 (2001) 213–219.
[12] Y. Perl, Y. Shiloach, Finding two disjoint paths between two pairs of vertices in a graph, J. ACM 25 (1) (1978) 1–9.
[13] D. Peleg, E. Upfal, Constructing disjoint paths on expander graphs, Combinatorica 9 (3) (1989) 289–313.
[14] D. Ronen, Y. Perl, Heuristics for finding a maximum number of disjoint bounded paths, Networks 14 (4) (1984) 531–544.
[15] J.W. Suuballe, Disjoint paths in a network, Networks 4 (2) (1974) 125–145.
[16] Sartaj K. Sahni, Algorithms for scheduling independent tasks, J. ACM 23 (1976) 116–127.
[17] Sartaj Sahni, General techniques for combinatorial approximation, Oper. Res. 25 (6) (1977) 920–936.
[18] Arthur Warburton, Approximation of Pareto optima in multiple-objective shortest path problems, Oper. Res. 35 (1987) 70–79.
[19] Bang Ye Wu, A note on approximating the min-max vertex disjoint paths on directed acyclic graphs, J. Comput. System Sci. 77 (6) (2011) 1054–1057.
[20] Chih-Chiang Yu, Chien-Hsin Lin, Biing-Feng Wang, Improved algorithms for finding length-bounded two vertex-disjoint paths in a planar graph and minmax k vertex-disjoint paths in a directed acyclic graph, J. Comput. System Sci. 76 (8) (2010) 697–708.