



Simple paths with exact and forbidden lengths

Alexandre Dolgui, Mikhail Kovalyov, Alain Quilliot

► To cite this version:

Alexandre Dolgui, Mikhail Kovalyov, Alain Quilliot. Simple paths with exact and forbidden lengths. Naval Research Logistics, 2018, 65 (1), pp.78 - 85. 10.1002/nav.21783 . hal-01774847

HAL Id: hal-01774847

<https://hal.science/hal-01774847v1>

Submitted on 24 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simple Paths with Exact and Forbidden Lengths

Alexander Dolgui^a, Mikhail Y. Kovalyov^{b,*}, Alain Quilliot^c

^a*IMT Atlantique, LS2N, CNRS, La Chantrierie, 4, rue Alfred Kastler - B.P. 20722, F-44307*

Nantes Cedex 3, France, E-mail: alexandre.dolgui@imt-atlantique.fr

^b*United Institute of Informatics Problems, National Academy of Sciences of Belarus, Minsk,*

Belarus, E-mail: kovalyov_my@newman.bas-net.by

^c*LIMOS, UMR CNRS 6158, Bat. ISIMA, Université Blaise Pascal, Campus des Cézeaux,*

BP 125, 63173 Aubiere, France, E-mail: alain.quilliot@isima.fr

Abstract

We study new decision and optimization problems of finding a simple path between two given vertices in an arc weighted directed multigraph such that the path length is equal to a given number or it does not fall into the given forbidden intervals (gaps). A fairly complete computational complexity classification is provided and exact and approximation algorithms are suggested.

Keywords: shortest path problem; longest path problem; exact path length; forbidden path length; computational complexity; approximation

1 Introduction

Let $G = (V, A)$ be an arbitrary *arc weighted directed multigraph*, which we further call *graph* and where $V = \{1, \dots, n\}$ is the set of *vertices* and A is the set of *arcs*, $|A| = m$. Arc $a_{ij}^{(r)} \in A$ is defined by its *head* vertex $i \in V$, *tail* vertex $j \in V$ and its *copy marker* r , if there are several arcs with the same head and the same tail. A *length* $l(a)$, which is an arbitrary integer number, is associated with each arc $a \in A$, and the length $L(P)$ of a *path* P from one specified vertex of G to another is the total length of its arcs. A path with no vertex repetition is called *simple*.

We study the following problems EXACT PATH(α), PATH GAPS, SHORT PATH GAPS and LONG PATH GAPS of finding a simple path from a given vertex s to a given vertex t in the graph G . Let integer numbers $\underline{f}_i, \bar{f}_i$, $i = 1, \dots, k$, be given such that $L_\Sigma^- \leq \underline{f}_1 \leq \bar{f}_1 < \underline{f}_2 \leq \bar{f}_2 < \dots < \underline{f}_k \leq \bar{f}_k \leq L_\Sigma^+$, where $L_\Sigma^- = \sum_{a \in A, l(a) \leq 0} l(a)$ is the total non-positive arc length

*Corresponding author

and $L_{\Sigma}^+ := \sum_{a \in A, l(a) > 0} l(a)$ is the total positive arc length. Here we assume that any sum is equal to zero if it is taken over an empty set. Denote $[\underline{f}, \bar{f}] = \{\underline{f}, \underline{f} + 1, \dots, \bar{f}\}$.

Problem EXACT PATH(α): Given number α , find a simple path P from vertex s to vertex t in the graph G such that its length $L(P) = \alpha$.

Problem PATH GAPS: Find a simple path P from vertex s to vertex t in the graph G such that the length $L(P) \notin \{[\underline{f}_1, \bar{f}_1], \dots, [\underline{f}_k, \bar{f}_k]\}$. We call intervals $[\underline{f}_i, \bar{f}_i]$, $i = 1, \dots, k$, *forbidden (path length) gaps*. The length of a path should not fall into these gaps.

We also study three special cases of the problem PATH GAPS: the case $k = 2$ of two forbidden gaps $[\underline{f}_1, \bar{f}_1]$ and $[\underline{f}_2, \bar{f}_2]$, the case $k = 1$ of a single forbidden gap $[\underline{f}_1, \bar{f}_1]$, and a sub-case of the latter case, in which a single path length α is forbidden, that is, $\underline{f}_1 = \bar{f}_1 = \alpha$. We denote these cases as PATH-2-GAPS, PATH-1-GAP and PATH NO(α), respectively.

Problem SHORT (LONG) PATH GAPS: Differs from PATH GAPS in that a *shortest (longest) simple path* with the length not from the forbidden gaps has to be found.

Occasionally, we will assume that graph G is undirected, which will be explicitly indicated. Problems PATH GAPS and SHORT PATH GAPS appear in route planning from one point of a network to another such that the arrival time does not fall into the given time intervals when the service required at the destination point is not available. This situation is similar to that in the vehicle routing problems with time windows, see, for example, Bräysy and Gendreau [6] for the formulation and a survey of the results for these problems. The problem LONG PATH GAPS can be used for modeling a situation, in which profits are collected over the route segments while traveling from one point of a network to another. If the total profit is less than a given value $B > 0$, then it is not worth traveling. The problem is to find a route such that the total profit is maximized and it does not fall into the forbidden gap $[0, B]$. It is clear that the forbidden gap constraint is redundant if the goal is to find an exact solution. However, the problem is NP-hard in general, and the gap constraint is essential for an approximate solution. Vehicle routing problems with profits have been studied by Archetti et al. [3]. In a situation of goods collection over segments of a path and their transportation in containers, there can be a requirement of container capacity utilization, leading to the forbidden gaps. For example, assume that the capacity of any container is 30 goods, and each container is required to be filled with at least 25 goods. A fixed number of goods associated with a path segment has to be collected if this segment is visited. In this situation, the total number of collected goods should fall into the intervals $[25, 30]$, $[50, 60]$, $[75, 90]$, $[100, 120]$,

$[125, \infty]$, and the forbidden intervals are $[0,24]$, $[31,49]$, $[61,74]$, $[91,99]$ and $[121,124]$. Feasible loads of the containers can be decided after a feasible path has been determined.

To the best of our knowledge, no literature exists on problems with the forbidden objective values gaps. The only exception is our recent paper [13] on a 0-1 knapsack problem with the forbidden objective function values. On the other hand, several *exact value* (or exact cost) combinatorial problems have been studied in the literature, which concern the existence of a solution with a given objective function value. The exact value assignment problem has been studied by Papadimitriou and Yannakakis [31] who proved its NP-completeness in the ordinary sense, and Karzanov [22] who developed a polynomial time algorithm for the case of 0-1 costs. Pseudo-polynomial time algorithms for the exact value spanning tree problem, the exact value perfect matching problem on planar graph, the exact value cycle sum problem on planar directed graph and the exact value cut problem on planar or toroidal graphs have been presented by Leclerc [25] and Barahona and Pulleyblank [4]. A number of computational complexity and algorithmic results for the exact weight (maximum) independent set problem on various classes of graphs have been obtained by Milanic and Monnot [30]. Computational complexity of the exact weight subgraph problems, in which the number of vertices of the subgraph is a constant, has been studied by Vassilevska and Williams [35] and Abboud and Lewi [1]. López et al. [27] studied the problem EXACT PATH(α), which they showed to be NP-complete and suggested modifications of the goal search (A^*) and bidirectional search algorithms for the solution. The problem EXACT PATH(α) can also be formulated as a special case of a *constrained path problem*, which is to maximize path length, subject to the constraint that the path length does not exceed a certain value, or to minimize the path length, subject to the constraint that the path length is at least a certain value. There exists a bulk of the literature on the constrained and bi-objective path problems, see, for example, Jokschi [20], Dial [12], Hansen [18], Aneja et al. [2], Desrochers [11], Warburton [36], Hassin [19], Lorenz and Raz [28], Ergun et al. [14], Righini and Salani [32], Boland et al. [5], Garcia [16], Tsaggouris and Zaroliagis [34] and Bruegem et al. [8]. We will present new observations about the relations between the problem EXACT PATH(α) and other path problems with the forbidden gaps.

The rest of the paper is organized as follows. In the next section, we describe connections of the new problems with the earlier studied problems that provide a fairly complete computational complexity classification of the new problems and some new algorithmic results.

Section 3 studies the case in which the graph is acyclic and the arc lengths are non-negative. While the problem EXACT PATH(α) and any of the problems PATH GAPS, SHORT PATH GAPS and LONG PATH GAPS with at least two forbidden gaps are NP-hard for this special case, an efficient approximation scheme is suggested, which delivers a solution with value close to the optimum but, possibly, violating a gap constraint with a given relative error. Polynomial time algorithms are presented for a more restrictive special case of these problems, in which forbidden gaps are polynomially bounded but the arc lengths are not. The paper concludes with a table of the obtained results and suggestions for future research.

2 Connections with the earlier studied problems

Observe that if graph G contains directed cycles, then the problems EXACT PATH(α) and PATH-1-GAP are NP-complete in the strong sense even if the arc lengths are all equal to one because the NP-complete problem HAMILTONIAN PATH (Garey and Johnson [17]) reduces to EXACT PATH(α) by setting $\alpha = n - 1$ and it reduces to PATH-1-GAP by setting $[\underline{f}_1, \bar{f}_1] = [1, n - 2]$.

We further abbreviate directed acyclic (multi)graph to *DAG*. Assume that G is a DAG. In this case, EXACT PATH(α) is NP-complete in the ordinary sense as it was mentioned by Lopéz et al. [27]. Furthermore, it is pseudo-polynomially solvable by the following folkloric dynamic programming algorithm, denoted as DP-ALL-LENGTHS. This algorithm scans vertices in a topological order (cf. Cormen et al. [10]) and constructs paths from vertex s to the successor vertices j . A *state* (j, f) is associated with a path from vertex s to vertex j , where f is the length of this path. If a complete path P^0 from vertex s to vertex t with length $L(P^0)$ goes via a vertex j and a sub-path of P^0 is in the state (j, f) , then any (incomplete) path in this state can be extended to a complete path P from s to t with the same length $L(P) = L(P^0)$.

Algorithm DP-ALL-LENGTHS recursively generates states (j, f) . Let $S(j)$ denote the set of states (j, f) generated for vertex j , which differ by the path lengths f . The initialization is $S(s) = \{(s, 0)\}$. Vertex s is labeled. In the general recursion step, a vertex j is identified whose predecessor vertices i are all labeled. Since G is a DAG, such a vertex always exists. Set $S(j)$ of states of the identified vertex is calculated as follows.

$$S(j) := \left\{ \left(j, f + l(a_{ij}^{(r)}) \right) \mid a_{ij}^{(r)} \in A, (i, f) \in S(i) \right\}.$$

After that, vertex j is labeled and the next vertex with all predecessor vertices labeled is identified. Ultimately, set $S(t)$ is generated, and the corresponding paths from s to t are found by backtracking. If a state $(t, \alpha) \in S(t)$, then the problem EXACT PATH(α) has a solution. Otherwise, it has no solution. The algorithm DP-ALL-LENGTHS can be implemented to run in $O(m(|L_{\Sigma}^{-}| + L_{\Sigma}^{+}))$ time.

Observe that the problem EXACT PATH(α) is a special case of the problem PATH-2-GAPS: an instance of EXACT PATH(α) is an instance of PATH-2-GAPS with gaps $[\underline{f}_1, \bar{f}_1] = [L_{\Sigma}^{-}, \alpha - 1]$ and $[\underline{f}_2, \bar{f}_2] = [\alpha + 1, L_{\Sigma}^{+}]$. Therefore, PATH-2-GAPS is NP-complete in the strong sense for graphs with directed cycles and unit arc lengths and it is NP-complete in the ordinary sense for DAGs with non-negative arc lengths. Furthermore, any instance of PATH-2-GAPS can be solved by solving the same instance of EXACT PATH(α) for all α which are not from the gaps. Therefore, PATH-2-GAPS is pseudo-polynomially solvable for DAGs, and it is polynomially solvable for DAGs with polynomially bounded absolute values of arc lengths. Since the problem PATH-2-GAPS is a special case of any of the problems PATH GAPS, SHORT PATH GAPS and LONG PATH GAPS, the latter problems are NP-hard in the strong sense for graphs with directed cycles and unit arc lengths and they are NP-hard in the ordinary sense for DAGs with non-negative arc lengths.

Let there be an algorithm with time complexity $O(T)$ for the problem EXACT PATH(α). By running this algorithm for $\alpha = L_{\Sigma}^{-}, L_{\Sigma}^{-} + 1, \dots, L_{\Sigma}^{+}$, any of the problems PATH GAPS, SHORT PATH GAPS and LONG PATH GAPS can be solved in $O(T(|L_{\Sigma}^{-}| + L_{\Sigma}^{+}))$ time. If G is a DAG, then, by scanning path lengths in the set $S(t)$ generated by the algorithm DP-ALL-LENGTHS, and selecting appropriate lengths and corresponding paths, all the problems PATH GAPS, SHORT PATH GAPS and LONG PATH GAPS can be solved in $O(m(|L_{\Sigma}^{-}| + L_{\Sigma}^{+}))$ time. Below more results are established for the problems PATH-1-GAP and PATH NO(α).

Observation 1 *Problem PATH-1-GAP reduces to solving both classic shortest and longest path problems.*

Proof. Note that an instance of the problem PATH-1-GAP and the corresponding instance of any of the two above mentioned classic path problems have a solution only if a path from s to t exists. Assume that it is the case. Let L_{short} and L_{long} denote lengths of the shortest and longest simple paths, respectively, in the classic problems. If $[L_{short}, L_{long}] \subseteq [\underline{f}_1, \bar{f}_1]$, then the instance of the problem PATH-1-GAP has no solution. Otherwise, if $L_{short} < \underline{f}_1$,

then the shortest simple path is a solution of the instance of the problem PATH-1-GAP, and if $L_{long} > \bar{f}_1$, then the longest simple path is a solution of this instance. ■

Since both classic shortest path and longest path problems can be solved in $O(n + m)$ time for DAGs (cf. Cormen et al. [10]), the following corollary follows.

Corollary 1 *If G is a DAG, then the problem PATH-1-GAP, and hence, the problem PATH NO(α), can be solved in $O(n + m)$ time.*

Recently, several publications appeared that study the so-called NEXT-TO-SHORTEST PATH problem, which asks for a path from s to t in a graph G of the second shortest length. Lalgudi and Papaefthymiou [24] proved that this problem is strongly NP-complete if graph G contains directed cycles and arc lengths are non-negative. It also follows from their results that the problem is solvable in $O(n + m)$ time if G is a DAG in this case. Computational complexity is open if the graph contains directed cycles and the arc lengths are strictly positive. The latter case is solvable in $O(n^3)$ time for planar graphs, as shown by Wu and Wang [37]. If the graph is undirected, then the problem is polynomially solvable. For strictly positive edge lengths, algorithms with running times $O(n^3m)$, $O(n^3)$ and $O(n^2)$ were successively presented by Krasikov and Noble [23], Li et al. [26] and Kao et al. [21]. For non-negative edge lengths, an $O(n^6m)$ time algorithm is presented by Zhang and Nagamochi [38]. Below we will assume that the path in the NEXT-TO-SHORTEST PATH problem is required to be simple.

Observation 2 *Problem PATH NO(α) reduces to solving both the classic shortest path problem and the NEXT-TO-SHORTEST PATH problem.*

Proof. Similar to the proof of Observation 1, assume without loss of generality that a path from s to t exists. Let L_{short} and $L_{short} + \delta$, $\delta > 0$, denote lengths of the shortest and next-to-shortest simple paths, respectively. If $L_{short} \neq \alpha$, then the shortest simple path is a solution of the instance of the problem PATH NO(α). If $L_{short} = \alpha$, then the next-to-shortest simple path is a solution of this instance. ■

Corollary 2 *The following special cases of the problem PATH NO(α) are polynomially solvable:*

- *if graph G is directed and planar and arc lengths are strictly positive, then PATH NO(α) can be solved in $O(n^3)$ time [37];*

- if graph G is undirected and arc lengths are strictly positive, then $\text{PATH NO}(\alpha)$ can be solved in $O(n^2)$ time [21];
- if graph G is undirected and arc lengths are non-negative, then $\text{PATH NO}(\alpha)$ can be solved in $O(n^6m)$ time [38].

Let us now show that for graphs with directed cycles and non-negative arc lengths the problem $\text{PATH NO}(\alpha)$ is difficult.

Observation 3 *If graph G contains directed cycles and the arc lengths are all equal to 0 but one arc length is equal to 1, then any of the problems $\text{EXACT PATH}(\alpha)$, $\text{PATH NO}(\alpha)$, PATH-1-GAP , PATH-2-GAPS , PATH GAPS , SHORT PATH GAPS and LONG PATH GAPS is NP-complete in the strong sense.*

Proof. Fortune et al. [15] proved that the problem $\text{TWO DISJOINT PATHS}$ is NP-complete in the strong sense. In this problem, given vertices s_1, t_1, s_2 and t_2 of a directed graph with directed cycles, the question concerns the existence of a simple path from s_1 to t_1 and a simple path from s_2 to t_2 such that these two paths have no common vertex. It can easily be verified that an instance of $\text{TWO DISJOINT PATHS}$ has a solution if and only if $\text{EXACT PATH}(\alpha)$ for $\alpha = 1$ (respectively, $\text{PATH NO}(\alpha)$ for $\alpha = 0$) has a solution for the same graph as in $\text{TWO DISJOINT PATHS}$ but with an extra arc (t_1, s_2) whose length is equal to one and with all other arc lengths equal to zero. Therefore, $\text{EXACT PATH}(\alpha)$ for $\alpha = 1$ and $\text{PATH NO}(\alpha)$ for $\alpha = 0$ are NP-complete in the strong sense. The other problems mentioned in the observation are generalizations of either the problem $\text{PATH NO}(\alpha)$ or the problem $\text{EXACT PATH}(\alpha)$ (for PATH-2-GAPS), and therefore, they cannot be easier. ■

It is worth noting that the problem $\text{TWO DISJOINT PATHS}$ in an undirected graph is solvable in almost linear time by the algorithm of Tholey [33].

3 DAGs with non-negative arc lengths

In this section, we assume that G is a DAG and the arc lengths are non-negative integer numbers. The latter assumption implies $L_{\Sigma}^- = 0$. For this special case, the problem $\text{EXACT PATH}(\alpha)$ and any of the problems PATH GAPS , SHORT PATH GAPS and LONG PATH GAPS with at least two forbidden gaps are NP-hard, as it is indicated in Section 2. In the following

sub-section, an approximation scheme is described for this special case, which delivers a solution with value close to the optimum with a given relative error ε , but, perhaps, inside a forbidden gap. In Sub-section 3.2, we suggest polynomial time algorithms for a more restrictive special case of these problems, in which forbidden gaps values are polynomially bounded but the arc lengths are not.

3.1 Approximation

Let ε be an arbitrary given number such that $0 < \varepsilon \leq 1$. We first suggest a family of algorithms (approximation scheme), each of which is specified by ε and a positive integer number β , and is denoted as $DP_{\varepsilon, \beta}$, such that, for any instance of the problem EXACT PATH(α) with $\alpha \in \{0, 1, \dots, \beta\}$, which has a solution, algorithm $DP_{\varepsilon, \beta}$ finds in $O(\frac{m}{\varepsilon})$ time a (simple) path from s to t in DAG G with non-negative integral arc lengths, whose length $F^{(\varepsilon, \alpha, \beta)}$ satisfies relations $F^{(\varepsilon, \alpha, \beta)} \leq \beta$ and $|\alpha - F^{(\varepsilon, \alpha, \beta)}| \leq \varepsilon\beta$.

Algorithm $DP_{\varepsilon, \beta}$ is a modification of the algorithm DP-ALL-LENGTHS. The modification concerns generation of the sets of states $S(j)$. After the set

$$S(j) = \left\{ (j, f + l(a_{ij}^{(r)})) \mid a_{ij}^{(r)} \in A, (i, f) \in S(i) \right\}$$

has been produced, states (j, f) with path lengths $f \geq \beta + 1$ are excluded from it and set $S^{(1)}(j) = \{(j, f) \mid (j, f) \in S(j), f \leq \beta\}$ is generated. Further, the set $S^{(1)}(j)$ is partitioned into disjoint subsets $X^{(1)}(j), X^{(2)}(j), \dots, X^{(u)}(j)$ such that $|f_1 - f_2| \leq \varepsilon\beta$ for any states (f_1, j) and (f_2, j) from the same subset. This partitioning can be done in $O(|S^{(1)}(j)|)$ time by calculating value $\lfloor f/(\varepsilon\beta) \rfloor$ for each $(f, j) \in S^{(1)}(j)$ and assigning states with the same value $\lfloor f/(\varepsilon\beta) \rfloor$ to the same subset $X^{(h)}(j)$ such that $h = \lfloor f/(\varepsilon\beta) \rfloor$. The number of the subsets does not exceed $O(1/\varepsilon)$. Since empty subsets are of no interest, they are removed and the non-empty subsets are re-numbered $X^{(1)}(j), X^{(2)}(j), \dots, X^{(u)}(j)$.

In each non-empty subset $X^{(h)}(j)$, minimum and maximum numbers, $f_{\min}^{(h, j)}$ and $f_{\max}^{(h, j)}$, which are the same number if $|X^{(h)}(j)| = 1$, are selected, $h = 1, \dots, u$. Thus, for any state $(j, f) \in S^{(1)}(j)$, there is an index $h \in \{1, \dots, u\}$ such that $f_{\min}^{(h, j)} \leq f \leq f_{\max}^{(h, j)}$ and $f_{\max}^{(h, j)} - f_{\min}^{(h, j)} \leq \varepsilon\beta$.

Finally, set $S^{(2)}(j) = \{(j, f_{\min}^{(h, j)}), (j, f_{\max}^{(h, j)}) \mid h = 1, \dots, u\}$ is generated, original set $S(j)$ is updated such that $S(j) := S^{(2)}(j)$, vertex j is labeled, and the next vertex with all predecessor vertices labeled is identified. We have $|S^{(2)}(j)| \leq 2u = O(1/\varepsilon)$. Therefore, $|S^{(1)}(j)| \leq$

$\sum_{i \in K_j} |S_i^{(2)}| = O(|K_j|/\varepsilon)$, where K_j is the set of vertices immediately preceding j in G , and $|K_j|$ is the *indegree* of j , for any $j \in V$.

Recall that the running time of the optimal algorithm DP-ALL-LENGTHS is $O(m(|L_\Sigma^-| + L_\Sigma^+))$, where $m = \sum_{j \in V} |K_j|$ and $O(|L_\Sigma^-| + L_\Sigma^+)$ is an upper bound on the number of distinct path lengths. Similarly, the running time of the approximation algorithm $DP_{\varepsilon, \beta}$ can be evaluated as $O(\sum_{j \in V} |K_j|U)$, where U is an upper bound on the number of distinct “rounded” path lengths associated with the same vertex. We have $U \leq O(1/\varepsilon)$, therefore, $DP_{\varepsilon, \beta}$ can be implemented to run in $O(\frac{m}{\varepsilon})$ time.

We next prove an important property of the algorithm $DP_{\varepsilon, \beta}$.

Theorem 1 *If problem EXACT PATH(α), $\alpha \leq \beta$, has solution, then there exists a state $(t, F^{(\varepsilon, \alpha, \beta)}) \in S(t)$ in the algorithm $DP_{\varepsilon, \beta}$ and the corresponding path with length $F^{(\varepsilon, \alpha, \beta)}$ such that $F^{(\varepsilon, \alpha, \beta)} \leq \beta$ and $|\alpha - F^{(\varepsilon, \alpha, \beta)}| \leq \varepsilon\beta$.*

Proof. Let path (j_1, \dots, j_r) be a solution of EXACT PATH(α), where $j_1 = s$ and $j_r = t$. The proof can be given by an induction on j_i . Assume that, for a state (j_i, f) in DP-ALL-LENGTHS, $1 \leq i \leq r-1$, preceding the final state (t, α) , there exists a state (j_i, f_1) in the algorithm $DP_{\varepsilon, \beta}$ such that $f_1 \leq f \leq \beta$ and $|f_1 - f| \leq \varepsilon\beta$. This assumption is satisfied for $j_i = s$. Then there exist states $(j_i, f_{\min}^{(h, j_i)})$ and $(j_i, f_{\max}^{(h, j_i)})$ in $DP_{\varepsilon, \beta}$ such that $f_{\min}^{(h, j_i)} \leq f_1 \leq f_{\max}^{(h, j_i)} \leq \beta$ and $f_{\max}^{(h, j_i)} - f_{\min}^{(h, j_i)} \leq \varepsilon\beta$. These two relations, together with $f_1 \leq f \leq \beta$ and $|f_1 - f| \leq \varepsilon\beta$, imply that for either $f_2 = f_{\min}^{(h, j_i)}$ or $f_2 = f_{\max}^{(h, j_i)}$ we have $f_2 \leq f \leq \beta$ and $|f_2 - f| \leq \varepsilon\beta$. The state (j_i, f_2) is generated in $DP_{\varepsilon, \beta}$. Further, if state (j_i, f) is extended to a state $(j_{i+1}, f + l)$, preceding the final state (t, α) , in DP-ALL-LENGTHS, then the state (j_i, f_2) is extended to the state $(j_{i+1}, f_2 + l)$ in $DP_{\varepsilon, \beta}$ such that $f_2 + l \leq f + l$ and $|(f_2 + l) - (f + l)| \leq \varepsilon\beta$. Repetition of this inductive argument for all vertices j_1, \dots, j_r completes the proof. \blacksquare

We now describe our ultimate approximation scheme $\{E_\varepsilon\}$. For any given ε , $0 < \varepsilon \leq 1$, algorithm E_ε consists of applications of the algorithm $DP_{\varepsilon/2, \beta}$ for $\beta \in W := \{0, \lfloor (1 + \varepsilon/2) \rfloor, \lfloor (1 + \varepsilon/2)^2 \rfloor, \dots, \lfloor (1 + \varepsilon/2)^w \rfloor, L_\Sigma^+\}$, where w is defined to satisfy $(1 + \varepsilon/2)^w < L_\Sigma^+$ and $(1 + \varepsilon/2)^{w+1} \geq L_\Sigma^+$. By taking logarithm with base two from both sides of the latter relation and assuming that w is a real number, we obtain that this relation is satisfied for $w = (\log_2 L_\Sigma^+)/\log_2(1 + \varepsilon/2) - 1$. Since $\log_2(1 + \varepsilon/2) \geq \varepsilon/2$ for $0 < \varepsilon \leq 2$, we know that $w \leq 2(\log_2 L_\Sigma^+)/\varepsilon - 1$. Thus, $|W| \leq 2(\log_2 L_\Sigma^+)/\varepsilon$, and algorithm E_ε runs in $O(\frac{m}{\varepsilon^2} \log_2 L_\Sigma^+)$

time.

The following theorem establishes properties of solutions delivered by the algorithm E_ε .

Theorem 2 *For any instance of any of the problems PATH GAPS, SHORT PATH GAPS and LONG PATH GAPS on a DAG with non-negative arc lengths that has a solution, algorithm E_ε finds a solution (path from s to t), whose length $F^{(\varepsilon)}$ satisfies $|F^{(\varepsilon)} - F| \leq \varepsilon F$, where $F \in \{F^0, F^*\}$, F^0 is the value of any feasible solution of the problem PATH GAPS and F^* is the optimal value in the problems SHORT PATH GAPS and LONG PATH GAPS.*

Proof. Observe that, for any value α , $0 \leq \alpha \leq L_\Sigma^+$, there exists a value $\beta \in W$ such that $\alpha \leq \beta \leq \alpha(1 + \varepsilon/2)$. By Theorem 1, for these α and β algorithm $DP_{\varepsilon/2, \beta}$ will find a solution with value $F^{(\varepsilon/2, \alpha, \beta)}$ for the problem EXACT PATH(α), which satisfies $|F^{(\varepsilon/2, \alpha, \beta)} - \alpha| \leq \varepsilon\beta/2$. Taking into account $\beta \leq \alpha(1 + \varepsilon/2)$ and $0 < \varepsilon \leq 1$, we obtain $|F^{(\varepsilon/2, \alpha, \beta)} - \alpha| \leq \varepsilon(1 + \varepsilon/2)\alpha/2 \leq \varepsilon\alpha$. By substituting α with F , we see that the statement of the theorem is satisfied by setting $F^{(\varepsilon)} = F^{(\varepsilon/2, F, \beta^0)}$, where β^0 is the number from the set W such that $F \leq \beta^0 \leq F(1 + \varepsilon/2)$. ■

It follows from Theorem 2 and the definition of the absolute value that if there exists a feasible solution of the problem PATH GAPS or an optimal solution of any of the problems SHORT PATH GAPS and LONG PATH GAPS with value F , $F \in \{F^0, F^*\}$, such that $\bar{f}_i + 1 \leq F \leq \underline{f}_{i+1} - 1$, $i \in \{1, \dots, k\}$, then

$$(1 - \varepsilon)(\bar{f}_i + 1) \leq (1 - \varepsilon)F \leq F^{(\varepsilon)} \leq F(1 + \varepsilon) \leq (1 + \varepsilon)(\underline{f}_{i+1} - 1),$$

which means that ε is a guaranteed relative error of the gap constraint violation. This type of approximation with respect to the bounding constraints has been used, for example, by Brucker et al. [7] and Cheng et al. [9] for NP-hard scheduling problems.

By a similar argument, if the optimal value F^* of any of the problems SHORT PATH GAPS and LONG PATH GAPS is at least εF^* units away from any forbidden value, then the algorithm E_ε is guaranteed to find a feasible solution, in which case $\{E_\varepsilon\}$ is a *Fully Polynomial Time Approximation Scheme (FPTAS)*, see, for example, FPTASs of Hansen [18], Hassin [19], Lorenz and Raz [28] and Tsaggouris and Zaroliagis [34] for the constrained and multi-objective shortest path problems. The two-stage approach (algorithm $DP_{\varepsilon, \beta}$ first and then its application for $\beta \in W$) makes the approximation scheme $\{E_\varepsilon\}$ different from the existing FPTASs.

Note that recognizing the fact that the optimal value F^* is at least εF^* units away from any forbidden value for the problem SHORT PATH GAPS with two forbidden intervals $[0, \alpha - \varepsilon\alpha]$ and $[\alpha + \varepsilon\alpha, L_\Sigma^+]$ is as difficult as the problem EXACT PATH(α), which is NP-complete, see Section 2.

The results of this sub-section are summarized in the following theorem.

Theorem 3 *If G is a DAG and arc lengths are non-negative, then problems PATH GAPS, SHORT PATH GAPS and LONG PATH GAPS possess an approximation scheme $\{E_\varepsilon\}$ with running time $O(\frac{m}{\varepsilon^2} \log_2 L_\Sigma^+)$, which finds a solution, possibly infeasible, with any given relative error ε with respect to the optimal objective value and the gap constraints.*

The type of approximation in this sub-section is related to the concept of *resource augmentation*, which is used in the analysis of approximation algorithms, see Lucarelli et al. [29]. According to this concept, an approximation algorithm is allowed to find a solution in the domain which is broader than the feasible domain of the original problem.

3.2 Polynomially bounded forbidden path lengths

Let I denote the input length of any of the problems PATH GAPS, SHORT PATH GAPS and LONG PATH GAPS in binary encoding, $I = O(\sum_{a \in A} \log_2 |l(a)| + \sum_{i=1}^k (\log_2 |\bar{f}_i| + \log_2 |\underline{f}_i|))$, and let P_I denote a polynomial of I . If the absolute values $|l(a)|$ of the arc lengths are bounded by the polynomial P_I , then the problems PATH GAPS, SHORT PATH GAPS and LONG PATH GAPS are solvable by the algorithm DP-ALL-LENGTHS in $O(m^2 P_I)$ time for DAGs, because $L_\Sigma^- = 0$ and $L_\Sigma^+ \leq m P_I$ in this case.

In this sub-section, we study the case in which $\bar{f}_k \leq P_I$. We stress that the arc lengths are not assumed to be polynomially bounded. This case cannot be solved by a direct application of the algorithm DP-ALL-LENGTHS. Let us call an arc *short* if its length does not exceed P_I . Otherwise, an arc is called *long*. Recall that the considered graph is a DAG and the arc lengths are non-negative.

First of all, we apply the Breadth-First-Search algorithm (cf. Cormen et al. [10]) to modify the original graph in $O(n + m)$ time so that every arc belongs to at least one path from s to t . If the new graph contains no long arcs, then the problems PATH GAPS, SHORT PATH GAPS and LONG PATH GAPS can be solved by the algorithm DP-ALL-LENGTHS in $O(m^2 P_I)$ time, because $L_\Sigma^- = 0$ and $L_\Sigma^+ \leq m P_I$. Assume that the new graph contains at least

one long arc. In this case, an optimal solution of the classic longest path problem is an optimal solution for the problem LONG PATH GAPS. Indeed, either the former solution includes a long arc, and therefore, it is feasible with respect to the gaps, or its length is at least the length of a long arc, and again, it is feasible with respect to the gaps. Recall that a long arc is present in the new graph and consider now any of the problems PATH GAPS and SHORT PATH GAPS. There are two sub-cases to consider with respect to a feasible (respectively, optimal) solution of this problem: 1) this solution includes no long arc, and 2) it includes at least one long arc. For the sub-case 1), remove long arcs from the graph and apply algorithm DP-ALL-LENGTHS to find a feasible (for PATH GAPS) or optimal (for SHORT PATH GAPS) solution in $O(m^2 P_I)$ time. For the case 2), solve at most m classic shortest path problems, each of which is specified by a long arc a . The length of this arc is re-set to be a sufficiently small number, for example, $l(a) := -L_\Sigma^+$. In this case, any shortest path will necessarily go via the arc a . Denote such a path as P_a . Since arc a is originally long, path P_a is feasible with respect to the gaps. Feasible solution for the problem PATH GAPS or optimal solution for the problem SHORT PATH GAPS is the best solution with respect to the original arc lengths among at most $m + 1$ solutions: one solution for the sub-case 1) and at most m solutions for the sub-case 2). It can be found in $O(m^2 P_I + mn)$ time.

Thus, the case with non-negative arc lengths and $\bar{f}_k \leq P_I$ is as easy as the case with polynomially bounded absolute values of the arc lengths, and the following theorem holds.

Theorem 4 *If G is a DAG, arc lengths are non-negative and forbidden path lengths are polynomially bounded such that $\bar{f}_k \leq P_I$, then the problem LONG PATH GAPS is solvable in $O(m^2 P_I)$ time and any of the problems PATH GAPS and SHORT PATH GAPS are solvable in $O(m^2 P_I + mn)$ time.*

4 Conclusions and suggestions for future research

The main results of this paper are summarized in Table 1. There, “NP-h” and “sNP-h” abbreviate “NP-hard” and “strongly NP-hard”, and “SPP” and “LPP” abbreviate “shortest path problem” and “longest path problem”.

In the future, it is interesting to study path problems with exact and forbidden lengths for various specific graph classes.

Problem	Additional characteristics	Result	Reference
EXACT PATH(α), PATH-1-GAP, PATH-2-GAPS	G contains directed cycles, arc lengths are unit	sNP-h	HAMILTONIAN PATH
EXACT PATH(α), PATH NO(α), PATH-2-GAPS	G contains directed cycles, arc lengths are zero but one arc length is unit	sNP-h	TWO DISJOINT PATHS, Observation 3
PATH NO(α)	G contains directed cycles, strictly positive arc lengths	Open	
PATH GAPS, SHORT PATH GAPS LONG PATH GAPS		Reduction to EXACT PATH(α)	Enumeration on α , Section 2
PATH-1-GAP		Reduction to SPP & LPP	Observation 1
PATH-1-GAP	G is DAG	$O(n + m)$	Corollary 1
PATH NO(α)		Reduction to SPP & NEXT-TO-SHORTEST PATH	Observation 2
PATH NO(α)	G is planar, strictly positive arc lengths	$O(n^3)$	Corollary 2
EXACT PATH(α), $\forall \alpha$, PATH GAPS, SHORT PATH GAPS LONG PATH GAPS	G is DAG	$O(m(L_{\Sigma}^{-} + L_{\Sigma}^{+}))$	DP-ALL-LENGTHS, Section 2
EXACT PATH(α), PATH-2-GAPS	G is DAG, non-negative arc lengths	NP-h	PARTITION, [27]
SHORT PATH GAPS, LONG PATH GAPS	G is DAG, non-negative arc lengths	$O(\frac{m}{\epsilon^2} \log_2 L_{\Sigma}^{+})$ (solution can be gap infeasible)	Theorem 3
PATH GAPS, SHORT PATH GAPS	G is DAG, arc lengths are non-negative, $\bar{f}_k \leq P_I$	$O(m^2 P_I + mn)$	Sub-section 3.2
LONG PATH GAPS	G is DAG, arc lengths are non-negative $\bar{f}_k \leq P_I$	$O(m^2 P_I)$	Sub-section 3.2

Table 1: Computational complexity and algorithms

References

- [1] A. Abboud, K. Lewi, Exact weight subgraphs and the k-sum conjecture, In F.V. Fomin et al. (Eds.), ICALP 2013, Part I, LNCS 7965, pp. 1-12, Springer-Verlag Berlin Heidelberg, 2013.
- [2] Y.P. Aneja, V. Aggarwal, K.P.K. Nair, Shortest chain subject to side constraints, Naval Research Logistics Quarterly, 13 (1983) 295–302.
- [3] C. Archetti, M.G. Speranza, D. Vigo, Chapter 10: Vehicle routing problems with profits, In P. Toth, D. Vigo (Eds.) Vehicle Routing. Problems, Methods, and Applications, Second Edition, MOS-SIAM Series on Optimization, pp. 273–297, SIAM, 2014.
- [4] F. Barahona, W.R. Pulleyblank, Exact aborescences, matchings and cycles, Discrete Applied Mathematics 16 (1987) 91–99.
- [5] N. Boland, J. Dethridge, I. Dumitrescu, Accelerated label setting algorithms for the elementary resource constrained shortest path problem, Operations Research Letters 34

- (2006) 58–68.
- [6] O. Bräysy, M. Gendreau, Vehicle routing problem with time windows, part I: Route construction and local search algorithms, *Transportation Science* 39(1) (2005) 104–118.
 - [7] P. Brucker, M.Y. Kovalyov, Y.M. Shafransky, F. Werner, Batch scheduling with deadlines on parallel machines, *Annals of Operations Research* 83 (1998) 23–40.
 - [8] T. Breugem, T. Dollevoet, W. van den Heuvel, Analysis of FPTASes for the multi-objective shortest path problem, *Computers and Operations Research* 78 (2017) 44–58.
 - [9] T.C.E. Cheng, A. Janiak, M.Y. Kovalyov, Bicriterion single machine scheduling with resource dependent processing times, *SIAM Journal on Optimization* 8(2) (1998) 617–630.
 - [10] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to algorithms* (3rd ed.), MIT Press and McGraw-Hill, London, 2009.
 - [11] M. Desrochers, *La fabrication d’horaires de travail pour les conducteurs d’autobus par une methode de generation de colonnes*, Ph.D Thesis, Centre De Recherche Sur Les Transports, Publication 470, Universite de Montreal, Canada, 1986.
 - [12] R.B. Dial, A model and algorithm for multicriteria route-mode choice, *Transportation Research Part B* 13 (1979) 311–316.
 - [13] A. Dolgui, M.Y. Kovalyov, A. Quilliot, Knapsack problem with objective value gaps, *Optimization Letters* 11 (2017) 31–39.
 - [14] F. Ergun, R. Sinha, L. Zhang, An improved FPTAS for the restricted shortest path problem, *Information Processing Letters* 83 (2002) 287–291.
 - [15] S. Fortune, J.E. Hopcroft, J. Wyllie, The directed subgraph homeomorphism problem, *Theoretical Computer Science* 10 (1980) 111–121.
 - [16] R. Garcia, *Resource constrained shortest paths and extensions*, PhD thesis, Georgia Institute of Technology, 2009.
 - [17] M.R. Garey, D.S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, W.H. Freeman and Co., San Francisco, 1979.

- [18] P. Hansen, Bicriterion path problems, in: Theory and applications. Proc. 3rd Conf. Multiple Criteria Decision Making, Lecture Notes in Economics and Mathematical Systems, vol. 117, Springer, Berlin, 1979, pp. 109–127.
- [19] R. Hassin, Approximation schemes for the restricted shortest path problem, *Mathematics of Operations Research* 17 (1992) 36–42.
- [20] H.C. Joksche, The shortest route problem with constraints, *Journal of Mathematical Analysis and Applications* 14(2) (1966), 191–197.
- [21] K.-H. Kao, J.-M. Chang, Y.-L. Wang, J.S.-T. Juan, A quadratic algorithm for finding next-to-shortest paths in graphs, *Algorithmica* 61 (2011) 402–418.
- [22] A.V. Karzanov, Maximum matching of given weight in complete and complete bipartite graphs, *Cybernetics* 23 (1987) 8–13. Translation from *Kibernetika* 1 (1987) 7–11.
- [23] I. Krasikov, S.D. Noble, Finding next-to-shortest paths in a graph, *Information Processing Letters* 92 (2004) 117–119.
- [24] K.N. Lalgudi, M.C. Papaefthymiou, Computing strictly-second shortest paths, *Information Processing Letters* 63 (1997) 177–181.
- [25] Leclerc, M.: Polynomial time algorithms for exact matching problems, Masters Thesis, University of Waterloo, Waterloo, 1986.
- [26] S. Li, G. Sun, G. Chen, Improved algorithm for finding next-to-shortest paths, *Information Processing Letters* 99 (2006) 192–194.
- [27] C.L. López, R. Stern, A. Felner, Solving the target-value search problem, *Proceedings of the Seventh Annual Symposium on Combinatorial Search (SoCS 2014)*, 2014, 202–203.
- [28] D. Lorenz, D. Raz, A simple efficient approximation scheme for the restricted shortest path problem, *Operations Research Letters* 28 (2001) 213–219.
- [29] G. Lucarelli, N.K. Thang, A. Srivastav, D. Trystram, Online nonpreemptive scheduling in a resource augmentation model based on duality. In: *Proc. 24th European Symposium on Algorithms*, 2016.

- [30] Milanic, M., Monnot, J.: The exact weighted independent set problem in perfect graphs and related graph classes, *Electronic Notes in Discrete Mathematics* 35 (2009) 317–322.
- [31] C.H. Papadimitriou, M. Yannakakis, The complexity of restricted spanning tree problems, *Journal of the ACM* 29 (1982) 285–309.
- [32] G. Righini, M. Salani, Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints, *Discrete Optimization* 3 (2006) 255–273.
- [33] T. Tholey, Solving the 2-disjoint paths problem in nearly linear time, in: *STACS 2004, Lecture Notes in Computer Science*, vol. 2996, Springer, Berlin, 2004, pp. 350–361.
- [34] G. Tsaggouris, C. Zaroliagis, Multiobjective optimization: Improved FPTAS for shortest paths and non-linear objectives with applications, *Theory of Computing Systems* 45(1) (2009) 162–186.
- [35] Vassilevska, V., Williams, R.: Finding, minimizing, and counting weighted subgraphs. In Mitzenmacher, M. (ed.), *STOC*, pp. 455–464, Association for Computing Machinery, 2009.
- [36] A. Warburton, Approximation of Pareto optima in multiple-objective, shortest path problems, *Operations Research* 35 (1987) 70–79.
- [37] B.Y. Wu, H.-L. Wang, The next-to-shortest path problem on directed graphs with positive edge weights, *Networks* 65(3) (2015) 205–211.
- [38] C. Zhang, H. Nagamochi, The next-to-shortest path in undirected graphs with nonnegative weights, *Proceedings of 18th Computing: The Australasian Theory Symposium (CATS)*, Melbourne, Australia, 2012, 13–19.