Multi-criteria approximation schemes for the resource constrained shortest path problem

Markó Horváth · Tamás Kis

Abstract In the resource constrained shortest path problem we are given a directed graph along with a source node and a destination node, and each arc has a cost and a vector of weights specifying its requirements from a set of resources with finite budget limits. A minimum cost source-destination path is sought such that the total consumption of the arcs from each resource does not exceed its budget limit. In the case of constant number of weight functions we give a fully polynomial time multi-criteria approximation scheme for the problem which returns a source-destination path of cost at most the optimum, however, the path may slightly violate the budget limits. On the negative side, we show that there does not exist a polynomial time multi-criteria approximation scheme for the problem if the number of weight functions is not a constant. The latter result applies to a broad class of problems as well, including the multi-dimensional knapsack, the multi-budgeted spanning tree, the multi-budgeted matroid basis and the multi-budgeted bipartite perfect matching problems.

Keywords multi-criteria approximation algorithms · resource constrained shortest path · multi-budgeted combinatorial optimization

Markó Horváth

Institute for Computer Science and Control, Hungarian Academy of Sciences, H1111 Budapest, Kende str. 13–17, Hungary
ORCID: 0000-0002-1062-6272

Tamás Kis

Institute for Computer Science and Control, Hungarian Academy of Sciences, H1111 Budapest, Kende str. 13–17, Hungary
Tel.: +36-1-279-6156
Fax: +36-1-466-7503
E-mail: tamas.kis@sztaki.mta.hu
ORCID: 0000-0002-2759-1264

1 Introduction

The RESOURCE CONSTRAINED SHORTEST PATH problem (RCSPP) is an extension of the familiar shortest path problem. Briefly stated, we are given a directed graph, where each arc has a cost, and specifies its requirements from a set of resources with finite budget limits. The goal is to find a minimum cost directed path from a specified source node to a specified destination node such that the total consumption of the arcs from each resource type does not exceed the budget limit of the resource. This problem is NP-hard [4].

In this paper we investigate multi-criteria approximation schemes for the RCSPP, where the budget limits can be slightly violated. The motivation for our study is the recent paper by Grandoni et al. [6] in which the budgeted version of a number of combinatorial optimization problems are discussed. The budgeted version of a combinatorial optimization problem $\min\{c(S) : S \in \mathcal{S}\}$ (or $\max\{c(S) : S \in \mathcal{S}\}$) — where $\mathcal{S} \subseteq 2^U$ is the set of feasible solutions for a given universe U, and $c : U \to \mathbb{Q}$ is the objective function with $c(S) = \sum_{u \in S} c(u)$ for each $S \in \mathcal{S}$ — has an additional set of k weight functions, that is, a vector $\mathbf{w} : U \to \mathbb{Q}_{\geq 0}^k$ on the elements of U, and for each $S \in \mathcal{S}$, $\mathbf{w}(S) = \sum_{u \in S} \mathbf{w}(u)$. Further on, there is a limit $\mathbf{L} \in \mathbb{Q}_{>0}^k$ on the total budget allowed. The k-*budgeted optimization problem* can then be formulated as

$$\min/\max c(S) \text{ subject to } S \in \mathcal{S} \text{ and } \mathbf{w}_i(S) \leq \mathbf{L}_i \text{ for all } 1 \leq i \leq k. \qquad (1)$$

The RCSPP corresponds to the minimization version of the k-BUDGETED s–t PATH problem where $\mathcal{S} = \mathcal{P}_{st}$, i.e., the set of all s–t paths of a directed graph $D = (V, U)$ with specified nodes $s, t \in V$. Note that a path is a sequence of directed arcs which do not visit the same node twice. If $\mathcal{S}$ is the set of bases of a given matroid with ground set U, we have the k-BUDGETED MATROID BASIS problem; specially, if $\mathcal{S}$ is the set of spanning trees of a given undirected graph $G = (V, U)$, we have the k-BUDGETED SPANNING TREE problem. In the case of k-BUDGETED (BIPARTITE) (PERFECT) MATCHING problem the solution set $\mathcal{S}$ consists of the (perfect) matchings of an undirected (bipartite) graph $G = (V, U)$.

An α-*approximation algorithm*, $\alpha \geq 1$, for an optimization problem Π is a polynomial time algorithm which finds an α-approximate solution S for Π, that is, $c(S) \geq c(S_{OPT})/\alpha$ if Π is a maximization problem, and $c(S) \leq \alpha c(S_{OPT})$ if Π is a minimization one, where S_{OPT} is an optimal solution for Π. A *polynomial time approximation scheme* (PTAS) for an optimization problem Π is a family of approximation algorithms $\{A_\varepsilon\}_{\varepsilon > 0}$ such that A_ε is an $(1 + \varepsilon)$-approximation algorithm for Π for any $\varepsilon > 0$. A *fully polynomial time approximation scheme* (FPTAS) for Π is a PTAS with $\{A_\varepsilon\}_{\varepsilon > 0}$ such that A_ε runs in polynomial time in $1/\varepsilon$ as well. A *multi-criteria* $(\alpha_0; \alpha_1, \ldots, \alpha_k)$-*approximation algorithm*, $\alpha_i \geq 1$, for a k-budgeted optimization problem (1) is an algorithm which finds an α_0-approximate solution S such that $\mathbf{w}_i(S) \leq \alpha_i \mathbf{L}_i$ for all $1 \leq i \leq k$. We emphasize that S can violate the budget limit $\mathbf{L}_i$ within the

given factor α_i, $1 \leq i \leq k$, however, its cost $c(S)$ guaranteed to be within α_0 of the cost of the optimal solution of (1) which satisfies the budget limits. A *multi-criteria approximation scheme* for a k-budgeted optimization problem Π contains a $(1 + \varepsilon; 1 + \varepsilon, \ldots, 1 + \varepsilon)$-approximation algorithm for Π for any $\varepsilon > 0$. Furthermore, if the approximation scheme consists of $(1; 1+\varepsilon, \ldots, 1+\varepsilon)$-approximation algorithms (the solution found must be super optimal), then it will be denoted by $(1; 1 + \varepsilon, \ldots, 1 + \varepsilon)$-PTAS or -FPTAS.

The MULTI-CONSTRAINED PATH (MCP) problem can be considered as the decision version of the RCSPP, i.e., the objective function can be omitted, and the question is whether there exists an s–t path that satisfies the budget limits. An ε-approximation algorithm, $0 < \varepsilon < 1$, for the k-MCP problem is a polynomial time algorithm which returns an s–t path P with $\mathbf{w}_i(P) \leq \mathbf{L}_i$, $1 \leq i \leq k$, whenever there is an s–t path P' such that $\mathbf{w}_i(P') \leq (1 - \varepsilon)\mathbf{L}_i$, $1 \leq i \leq k$ [11].

Related work In the case of a single budget ($k = 1$), Hassin [7] proposed the first $(1 + \varepsilon; 1)$-FPTAS for the RCSPP on acyclic graphs with time complexity $O(m(n^2/\varepsilon) \log(n/\varepsilon))$, where m and n denote the number of arcs and nodes of the given directed graph, respectively. Ergun et al. [2] gave another $(1+\varepsilon; 1)$-FPTAS with improved running time of $O(mn/\varepsilon)$. For general graphs Lorenz and Raz [9] proposed an $(1 + \varepsilon; 1)$-FPTAS with time complexity of $O(mn(\log \log n + 1/\varepsilon))$. Goel et al. [5] gave an $(1; 1 + \varepsilon)$-FPTAS with running time $O((m + n \log n)n/\varepsilon)$.

In the case of $2 \leq k = O(1)$, one can obtain a multi-criteria $(1 + \varepsilon; 1 + \varepsilon, \ldots, 1+\varepsilon)$-FPTAS for the k-BUDGETED s–t PATH (including the RCSPP) and the k-BUDGETED SPANNING TREE problem based on the general technique of Papadimitriou and Yannakakis [10], however, there exists no $(\alpha_0; \alpha_1, \ldots, \alpha_k)$-approximation algorithm with two or more α_i's equal to 1 for these problems, unless $P = NP$ (see Grandoni et al. [6]). Further on, Grandoni et al. [6] describe $(1; 1+\varepsilon, \ldots, 1+\varepsilon)$-PTASs for the k-BUDGETED SPANNING TREE and the k-BUDGETED MATROID BASIS problems, however, they do not provide such an algorithm for the RCSPP, which is one of the motivations for our work. They also provide a $(1 + \varepsilon; 1 + \varepsilon, \ldots, 1 + \varepsilon)$-PTAS for the k-BUDGETED BIPARTITE MATCHING problem. Another motivation is that the method of Papadimitriou and Yannakakis [10], and the results of Grandoni et al. [6] work only if the number of weight functions, k, is a constant.

Song and Sahni [11] describe ε-approximation algorithms for the k-MCP problem. For an overview of exact and approximation algorithms for the k-MCP and its variants we refer to [3].

Our results Firstly, we show that the RCSPP admits no multi-criteria PTAS if the number of weight functions, k, is part of the input (not a constant), unless $P = NP$. This statement is also true for some other k-budgeted optimization problems (Theorem 1). On the positive side, we provide a $(1; 1+\varepsilon, \ldots, 1+\varepsilon)$-FPTAS for the case of $k = O(1)$ (Theorem 2). Notice that a direct application of the method of Papadimitriou and Yannakakis [10] would give only an

$(1 + \varepsilon; 1 + \varepsilon, \ldots, 1 + \varepsilon)$-FPTAS. Our multi-criteria FPTAS is a dynamic programming algorithm, similar to the SPPP algorithm of Lorenz and Raz [9], with a combination of the rounding technique of Song and Sahni [11]. The time complexity of our multi-criteria approximation scheme is $O(m(n/\varepsilon)^k)$, which for $k = 1$ matches that of Ergun et al. [2], who provided an $(1 + \varepsilon; 1)$-FPTAS for the RCSPP restricted to acyclic graphs.

Theorem 1 *If $P \neq NP$, and the number of weight functions, k, is part of the input (not a constant), then there exists no polynomial time multi-criteria approximation scheme for the minimization, nor for the maximization version of the k-BUDGETED s–t PATH, the k-BUDGETED SPANNING TREE, the k-BUDGETED MATROID BASIS, and the k-BUDGETED BIPARTITE PERFECT MATCHING problems.*

Theorem 2 *If the number of weight functions, k, is a constant then there exists a fully polynomial time $(1; 1 + \varepsilon, \ldots, 1 + \varepsilon)$-approximation scheme for the RCSPP.*

We emphasize that our result is valid for general graphs with non-negative weights and arbitrary costs, however, cycles with negative total cost are not allowed.

2 Proof of Theorem 1

In this section we assume that the number of weight functions, k, is part of the input (not a constant). First, we prove that unless $P = NP$, there exists no polynomial time multi-criteria approximation scheme for the RCSPP (Theorem 3). Based on the proof, it is a routine to show that there exists no multi-criteria PTAS for the minimization version of the k-BUDGETED SPANNING TREE (thus for the k-BUDGETED MATROID BASIS) and the k-BUDGETED BIPARTITE PERFECT MATCHING problems.

Theorem 3 *If $P \neq NP$, and the number of weight functions, k, is part of the input (not a constant), then there exists no polynomial time multi-criteria approximation scheme for the RCSPP.*

Proof We give a PTAS-preserving reduction from the VERTEX COVER (VC) problem to the RCSPP to show that there is no polynomial time $(1 + \varepsilon; 2 - \varepsilon, \ldots, 2 - \varepsilon)$-approximation algorithm for the RCSPP with $0 < \varepsilon < 0.3606$, unless $P = NP$. An instance of VC is given by an undirected graph G, and a minimum size subset of nodes $C \subseteq V(G)$ is sought such that for each edge $(u, v) \in E(G)$, $1 \leq |\{u, v\} \cap C|$. Given a VC instance, we create an instance of the RCSPP with $k = |E(G)|$ and with directed graph D as follows. For each node $v_i \in V(G)$, $1 \leq i \leq n$, we add two distinct nodes x_i and y_i, and also node x_{n+1} to $V(D)$, and three arcs (x_i, x_{i+1}), (x_i, y_i) and (y_i, x_{i+1}) to $A(D)$. Let $s = x_1$, $t = x_{n+1}$. We create a cost function $c : A(D) \to \mathbb{Q}_{\geq 0}$ such that

$c(x_i, x_{i+1}) = 1$ and $c(x_i, y_i) = c(y_i, x_{i+1}) = 0$ for all $1 \leq i \leq n$. We create weights $\mathbf{w} : A(D) \to \mathbb{Q}_{\geq 0}^{|E(G)|}$ such that

$$\mathbf{w}_{i,j}(a) = \begin{cases} 1, \text{ if } a = (x_i, y_i) \text{ or } a = (x_j, y_j), \\ 0, \text{ otherwise} \end{cases}$$

for each $(v_i, v_j) \in E(G)$. We set the corresponding budget limit $\mathbf{L}_{i,j}$ to 1.

Consider the one-to-one correspondence between the node sets of G and the s–t paths in D, such that to a node set $C \subseteq V(G)$ we assign the s–t path $P[C]$ in D consisting of arcs $\{(x_i, x_{i+1}) : v_i \in C\} \cup \{(x_i, y_i), (y_i, x_{i+1}) : v_i \notin C\}$. It is clear that for a node set $C \subseteq V(G)$ and the corresponding path $P[C]$, $c(P[C]) = |C|$, moreover, we claim that C is a vertex cover if and only if $P[C]$ satisfies the budget limits. If C is a vertex cover, then for each edge $(v_i, v_j) \in E(G)$ we have $1 \leq |\{v_i, v_j\} \cap C|$, thus $|\{(x_i, y_i), (x_j, y_j)\} \cap P[C]| \leq 1$, therefore $\mathbf{w}_{i,j}(P[C]) \leq \mathbf{L}_{i,j}$. The opposite direction can be shown similarly.

Assume that we have an $(1 + \varepsilon; 2 - \varepsilon, \ldots, 2 - \varepsilon)$-approximation algorithm for the RCSPP with $0 < \varepsilon < 0.3606$. Applying this algorithm for the RCSPP instance, we can find in polynomial time an s–t path P such that $c(P) \leq (1+\varepsilon)c(P_{OPT})$ and $\mathbf{w}_{i,j}(P) \leq (2-\varepsilon)\mathbf{L}_{i,j}$, for all $(v_i, v_j) \in E(G)$, where P_{OPT} is an optimal solution for the RCSPP. On the one hand, $c(P) \leq (1+\varepsilon)c(P_{OPT}) = (1 + \varepsilon)|C_{OPT}|$, where C_{OPT} is an optimal solution for the VC. On the other hand, $\mathbf{w}_{i,j}(P) \leq 2 - \varepsilon < 2$, i.e., $\mathbf{w}_{i,j}(P) \leq 1$ holds for all $(v_i, v_j) \in E(G)$, thus the set $C \subseteq V(G)$ corresponding to P is a vertex cover. To sum up, by applying the approximation algorithm for the RCSPP instance, we can find in polynomial time a vertex cover C in G such that $|C| \leq (1 + \varepsilon)|C_{OPT}|$ which is impossible for $\varepsilon < 0.3606$, unless $P = NP$ (see Dinur and Safra [1]). $\qquad\square$

Now, we prove that the MULTI-DIMENSIONAL KNAPSACK problem (MDKP) [12] does not admit a polynomial time multi-criteria approximation scheme if the number of dimensions, k, is part of the input (Theorem 4). Recall that an instance of the MDKP is given by a set of items U, where each item $u \in U$ has a cost $c(u)$, and a weight $\mathbf{w}(u) \in \mathbb{Q}_{\geq 0}^k$, and there is a weight limit $\mathbf{L} \in \mathbb{Q}_{\geq 0}^k$. A subset of items $S \subseteq U$ of maximum $c(S)$ value is sought such that $\mathbf{w}(S) \leq \mathbf{L}$. Apparently, it is the budgeted version of a trivial maximization problem over all subsets of U.

By using similar techniques it is easy to prove that there exists no multi-criteria PTAS for the maximization version of the k-BUDGETED s–t PATH, k-BUDGETED SPANNING TREE, the k-BUDGETED MATROID BASIS and the k-BUDGETED BIPARTITE PERFECT MATCHING problems, if the number of weight functions, k, is part of the input (not a constant), unless $P = NP$.

Theorem 4 *If $P \neq NP$, and if the dimension k of the MDKP is part of the input (not a constant), then there does not exist a polynomial time multi-criteria approximation scheme for the MDKP.*

Proof We give a PTAS-preserving reduction from the INDEPENDENT SET (IS) problem to the MDKP. An instance of IS is given by an undirected graph

$G = (V, E)$, and a maximum size subset of nodes $S \subseteq V$ is sought such that for each edge $(u, v) \in E$, $|\{u, v\} \cap S| \leq 1$. Given an IS instance, we create an instance of the MDKP with $k = |E|$ as follows. We create a set of items $U = \{u_1, \ldots, u_n\}$ where item u_i corresponds to node $v_i \in V$ and it has a cost $c(u_i) = 1$. We create weights $\mathbf{w}_{i,j} : U \to \mathbb{Q}_{\geq 0}^{|E|}$ such that

$$\mathbf{w}_{i,j}(u) = \begin{cases} 1, & \text{if } u \text{ corresponds to } v_i \text{ or } v_j, \\ 0, & \text{otherwise} \end{cases}$$

for each $(v_i, v_j) \in E$. We set the corresponding budget limit $\mathbf{L}_{i,j}$ to 1.

Consider a node set $I = \{v_{i_1}, \ldots, v_{i_p}\} \subseteq V$ and the corresponding set of items $S = \{u_{i_1}, \ldots, u_{i_p}\}$. If I is independent, then for each edge $(v_i, v_j) \in E$ we have $|\{v_i, v_j\} \cap I| \leq 1$, thus $|\{u_i, u_j\} \cap S| \leq 1$, therefore $\mathbf{w}_{i,j}(S) \leq \mathbf{L}_{i,j}$, that is, S satisfies the budget limits, moreover $c(S) = |I|$. The opposite direction (that is, the node set corresponding to a set of items that satisfies the budget limits is independent) can be shown similarly.

Similarly to the previous proof, if we had a $(1+\varepsilon; 2-\varepsilon, \ldots, 2-\varepsilon)$-approximation algorithm for the MDKP with $0 < \varepsilon < 1$, we could find in polynomial time an independent set I in G such that $|I| \geq |I_{OPT}|/(1 + \varepsilon)$ which is impossible, unless $P = NP$ (see Hastad [8]). $\square$

3 Proof of Theorem 2

We give a $(1; 1 + \varepsilon, \ldots, 1 + \varepsilon)$-FPTAS for the RCSPP, where the number of weight functions, k, is a constant. Recall, that this problem can be formulated as

$$\min_{P \in \mathcal{P}_{st}} \{c(P) : \mathbf{w}_i(P) \leq \mathbf{L}_i, \ i = 1, \ldots, k\}. \tag{2}$$

For a given an $\varepsilon > 0$ we scale and round the weights, that is, we define a scale vector $\mathbf{\Delta} \in \mathbb{Q}_{>0}^k$ and scaled weights $\bar{\mathbf{w}} \in \mathbb{Q}_{>0}^k$ as follows: for all $i = 1, \ldots, k$ let $\mathbf{\Delta}_i := \varepsilon \mathbf{L}_i/(n - 1)$, and for each arc $a \in A$ let $\bar{\mathbf{w}}_i(a) := d_i(a)\mathbf{\Delta}_i$, where $d_i(a) = 1$ if $\mathbf{w}_i(a) = 0$, otherwise $d_i(a)$ is a positive integer such that $(d_i(a) - 1)\mathbf{\Delta}_i < \mathbf{w}_i(a) \leq d_i(a)\mathbf{\Delta}_i$ holds. Note that $0 < \bar{\mathbf{w}}_i(a)$ holds for each arc a and $i = 1, \ldots, k$. Consider the following, scaled problem:

$$\min_{P \in \mathcal{P}_{st}} \{c(P) : \bar{\mathbf{w}}_i(P) \leq (1 + \varepsilon)\mathbf{L}_i, \ i = 1, \ldots, k\}. \tag{3}$$

For any s–t path P and $i = 1, \ldots, k$ we have

$$\bar{\mathbf{w}}_i(P) = \sum_{a \in P} \bar{\mathbf{w}}_i(a) \leq \sum_{a \in P}(\mathbf{w}_i(a) + \mathbf{\Delta}_i) \leq \sum_{a \in P} \mathbf{w}_i(a) + (n-1)\mathbf{\Delta}_i = \mathbf{w}_i(P) + \varepsilon \mathbf{L}_i,$$

since P consists of at most $n - 1$ arcs. Thus, if P is a feasible solution for the original problem (2), i.e., $\mathbf{w}_i(P) \leq \mathbf{L}_i$ holds for all $i = 1, \ldots, k$, we have $\bar{\mathbf{w}}_i(P) \leq (1 + \varepsilon)\mathbf{L}_i$, $1 \leq i \leq k$, i.e., P is feasible for the scaled problem (3) as well. Moreover, since by definition $\mathbf{w}_i(a) \leq \bar{\mathbf{w}}_i(a)$ holds for all arcs a and $i = 1, \ldots, k$, thus for each feasible solution P for the scaled problem (3) we

have $\mathbf{w}_i(P) \leq \bar{\mathbf{w}}_i(P) \leq (1+\varepsilon)\mathbf{L}_i$, $1 \leq i \leq k$. These imply the following proposition.

Proposition 1 *If problem (2) has a feasible solution, then any optimal solution for problem (3) is a $(1; 1+\varepsilon, \ldots, 1+\varepsilon)$-approximate solution for (2).*

3.1 Dynamic programming algorithm

In the following we use element-wise operations for vectors. That is, the Hadamard product of vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Q}^k$ is the vector $\mathbf{a} \circ \mathbf{b} \in \mathbb{Q}^k$ with $(\mathbf{a} \circ \mathbf{b})_i = \mathbf{a}_i \mathbf{b}_i$, $1 \leq i \leq k$. The inverse of vector $\mathbf{a} \in \mathbb{Q}^k_{>0}$ is the vector $\mathbf{a}^{-1} \in \mathbb{Q}^k_{>0}$ with $(\mathbf{a}^{-1})_i = 1/\mathbf{a}_i$, $1 \leq i \leq k$. For k-dimension vectors $\mathbf{a}$ and $\mathbf{b}$ we write $\mathbf{a} \leq \mathbf{b}$ $(\mathbf{a} < \mathbf{b})$ if $\mathbf{a}_i \leq \mathbf{b}_i$ $(\mathbf{a}_i < \mathbf{b}_i)$ holds for all $i = 1, \ldots, k$.

A *pattern* is a vector $\boldsymbol{\eta} = (\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_k)$ where $\boldsymbol{\eta}_i$ $(i = 1, \ldots, k)$ is a nonnegative integer. Note, that for any path P, there is a pattern $\boldsymbol{\eta}$ with nonzero elements such that $\bar{\mathbf{w}}(P) = \boldsymbol{\eta} \circ \boldsymbol{\Delta}$. A pattern $\boldsymbol{\eta}$ is *feasible*, if $\boldsymbol{\eta} \circ \boldsymbol{\Delta} \leq (1+\varepsilon)\mathbf{L}$ holds.

Proposition 2 *The number of feasible patterns is $O((n/\varepsilon)^k)$.*

Proof For any feasible pattern $\boldsymbol{\eta}$, $0 \leq \boldsymbol{\eta}_i \leq (1+\varepsilon)\mathbf{L}_i/\boldsymbol{\Delta}_i = (n-1)(1+1/\varepsilon)$ holds for all $i = 1, \ldots, k$, thus the number of feasible patterns is at most $((n-1)(1+1/\varepsilon)+1)^k = O((n/\varepsilon)^k)$. $\square$

For a node v and pattern $\boldsymbol{\eta}$ let $\chi(v, \boldsymbol{\eta})$ denote the cost of the minimum cost s–v path P such that $\bar{\mathbf{w}}(P) \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta}$, if any. By this, $\chi(t, \lfloor (1+\varepsilon)\mathbf{L} \circ \boldsymbol{\Delta}^{-1} \rfloor)$ is the optimal solution value of (3). Let $H = (\boldsymbol{\eta}^1, \boldsymbol{\eta}^2, \ldots, \boldsymbol{\eta}^{|H|})$ denote the set of the feasible patterns, where patterns are partially ordered by the element-wise comparison, that is, if $\boldsymbol{\eta}^p \leq \boldsymbol{\eta}^q$ holds for patterns $\boldsymbol{\eta}^p$ and $\boldsymbol{\eta}^q$, then $p \leq q$. Clearly $\boldsymbol{\eta}^1 = (0, \ldots, 0)$.

The sketch of the algorithm can be seen in Algorithm 1. In the initialization phase for each pattern $\boldsymbol{\eta}$ we set $\chi(s, \boldsymbol{\eta})$ to zero and $\chi(v, \boldsymbol{\eta})$ to infinity $(v \neq s)$. We iterate over the partially ordered set of feasible patterns (note that according to the initialization, we can skip pattern $\boldsymbol{\eta}^1 = (0, \ldots, 0)$), and in each iteration we visit each node in the graph. For a given pattern $\boldsymbol{\eta}$ and node v we examine the incoming arcs of v. Let (u, v) be an impending arc. If $\bar{\mathbf{w}}_i(u, v) > \boldsymbol{\eta}_i \boldsymbol{\Delta}_i$ holds for some $1 \leq i \leq k$, then there is no s–v path containing arc (u, v) such that $\bar{\mathbf{w}}(P) \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta}$, thus we cannot update $\chi(v, \boldsymbol{\eta})$. Otherwise $\bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1} \leq \boldsymbol{\eta}$, and by definition $0 < \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1}$, thus $0 \leq \boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1} < \boldsymbol{\eta}$, i.e., pattern $\boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1}$ was already examined in a former iteration, that is, $\chi(u, \boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1})$ is already computed (and valid), so we can update $\chi(v, \boldsymbol{\eta})$. Finally, we return $\chi(t, \lfloor (1+\varepsilon)\mathbf{L} \circ \boldsymbol{\Delta}^{-1} \rfloor)$ which is the optimal solution value of (3).

In Theorem 5 we prove the correctness of the algorithm.

Theorem 5 *After Algorithm 1 terminates, for each node $v \in V$ and for each pattern $\boldsymbol{\eta}$, $\chi(v, \boldsymbol{\eta})$ is equal to the cost of the minimum cost s–v path P such that $\bar{\mathbf{w}}(P) \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta}$.*

Algorithm 1 Dynamic programming algorithm for (3)

1: $\chi(s, \boldsymbol{\eta}) \leftarrow 0 \ (\boldsymbol{\eta} \in H)$
2: $\chi(v, \boldsymbol{\eta}) \leftarrow \infty \ (v \neq s, \boldsymbol{\eta} \in H)$
3: **for** $\boldsymbol{\eta} = \boldsymbol{\eta}^2, \boldsymbol{\eta}^3, \dots, \boldsymbol{\eta}^{|H|}$ **do**
4:     **for** $v \in V$ **do**
5:         **for** $a \in \{(u, v) \in A : \ \bar{\mathbf{w}}(u, v) \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta}\}$ **do**
6:             $\chi(v, \boldsymbol{\eta}) \leftarrow \min\{\chi(v, \boldsymbol{\eta}), \chi(u, \boldsymbol{\eta} - \bar{\mathbf{w}}(a) \circ \boldsymbol{\Delta}^{-1}) + c(a)\}$
7:         **end for**
8:     **end for**
9: **end for**
10: **return** $\chi(t, (\lfloor (n-1)(1 + 1/\varepsilon) \rfloor, \dots, \lfloor (n-1)(1 + 1/\varepsilon) \rfloor))$

Proof Basically, we prove that after the algorithm terminates $\chi(v, \boldsymbol{\eta})$ is equal to the cost of the minimum cost s–v walk P (i.e., vertices may be repeated) such that $\bar{\mathbf{w}}(P) \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta}$. However, according to our assumptions the graph does not contain cycles with negative total cost or negative total weight, therefore a minimum cost s–t walk always comprises an s–t path of the same cost. To prove the former statement, it is sufficient to show that after a pattern $\boldsymbol{\eta}$ is examined (i.e., the corresponding iteration is performed):

a) for each node v, if $\chi(v, \boldsymbol{\eta})$ is not infinity, then it is equal to the cost of an s–v walk P such that $\bar{\mathbf{w}}(P) \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta}$.
b) for each node v, if there is an s–v walk with $\bar{\mathbf{w}}(P) \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta}$, then $\chi(v, \boldsymbol{\eta}) \leq c(P)$ holds.

We prove these statements by induction. Clearly, statements a) and b) are satisfied before the first iteration is performed (i.e., after the initialization).

To prove statement a) consider a moment when $\chi(v, \boldsymbol{\eta})$ is updated (line 6), that is $\chi(v, \boldsymbol{\eta}) = \chi(\boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1}) + c(u, v)$ for some $(u, v) \in A$ with $\bar{\mathbf{w}}(u, v) \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta}$. Since $0 < \bar{\mathbf{w}}(u, v)$ holds by definition, thus $0 \leq \boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1} < \boldsymbol{\eta}$, i.e., pattern $\boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1}$ was already examined in a former iteration, that is, $\chi(u, \boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1})$ is already computed. By inductive assumption, $\chi(u, \boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1})$ is equal to the cost of an s–u walk P with $\bar{\mathbf{w}}(P) \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta} - \bar{\mathbf{w}}(u, v)$, thus $\chi(v, \boldsymbol{\eta})$ is equal to the cost of the walk $P' = P \cup \{(u, v)\}$ with $\bar{\mathbf{w}}(P') \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta}$.

To prove statement b) consider the shortest s–v path P such that $\bar{\mathbf{w}}(P) \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta}$ holds, and let (u, v) be its last arc, i.e., $P = P' \cup (u, v)$ for an s–u path P'. Clearly, P' is the minimum cost s–u walk such that $\bar{\mathbf{w}}(P') \leq \boldsymbol{\eta} \circ \boldsymbol{\Delta} - \bar{\mathbf{w}}(u, v)$ holds. On the one hand, by inductive assumption, we have $\chi(v, \boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1}) \leq c(P')$, and on the other hand we compared $\chi(v, \boldsymbol{\eta})$ and $\chi(u, \boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1}) + c(u, v)$ in the iteration of pattern $\boldsymbol{\eta}$ (line 6), therefore we have $\chi(v, \boldsymbol{\eta}) \leq \chi(u, \boldsymbol{\eta} - \bar{\mathbf{w}}(u, v) \circ \boldsymbol{\Delta}^{-1}) + c(u, v) \leq c(P') + c(u, v) = c(P)$. □

According to Proposition 2 we have at most $O((n/\varepsilon)^k)$ iterations, and in each iteration each arc is examined once, thus the running time of the algorithm is $O(m(n/\varepsilon)^k)$.

4 Final remarks

Our positive and negative results, along with the observations of Grandoni et al. [6] give a complete picture on the approximability of the RCSPP in terms of approximation schemes. However, a major open question is whether a $(1; 1 + \varepsilon, \ldots, 1 + \varepsilon)$-FPTAS exists for spanning trees in the case of constant number of weight functions.

References

1. I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of mathematics*, pages 439–485, 2005.
2. F. Ergun, R. Sinha, and L. Zhang. An improved FPTAS for restricted shortest path. *Information Processing Letters*, 83(5):287–291, 2002.
3. R. G. Garroppo, S. Giordano, and L. Tavanti. A survey on multi-constrained optimal path computation: Exact and approximate algorithms. *Computer Networks*, 54(17):3081–3107, 2010.
4. M. R. Gary and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman and Company, New York, 1979.
5. A. Goel, K. G. Ramakrishnan, D. Kataria, and D. Logothetis. Efficient computation of delay-sensitive routes from one source to all destinations. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 854–858. IEEE, 2001.
6. F. Grandoni, R. Ravi, M. Singh, and R. Zenklusen. New approaches to multi-objective optimization. *Mathematical Programming*, 146(1-2):525–554, 2014.
7. R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations research*, 17(1):36–42, 1992.
8. J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 627–636. IEEE, 1996.
9. D. H. Lorenz and D. Raz. A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 28(5): 213–219, 2001.
10. C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 86–92. IEEE, 2000.

11. M. Song and S. Sahni. Approximation algorithms for multiconstrained quality-of-service routing. *IEEE Transactions on computers*, 55(5):603–617, 2006.
12. H. M. Weingartner and D. N. Ness. Methods for the solution of the multidimensional 0/1 knapsack problem. *Operations Research*, 15(1):83–103, 1967.