THE VPN PROBLEM WITH CONCAVE COSTS

SAMUEL FIORINI, GIANPAOLO ORIOLO, LAURA SANITÀ, AND DIRK OLIVER THEIS

ABSTRACT. Only recently Goyal, Olver and Shepherd (Proc. STOC, 2008) proved that the symmetric Virtual Private Network Design (sVPN) problem has the tree routing property, namely, that there always exists an optimal solution to the problem whose support is a tree. Combining this with previous results by Fingerhut, Suri and Turner (J. Alg., 1997) and Gupta, Kleinberg, Kumar, Rastogi and Yener (Proc. STOC, 2001), sVPN can be solved in polynomial time.

In this paper we investigate a NP-hard generalization of sVPN, where the contribution of each edge to the total cost is proportional to some concave, non-decreasing function of the capacity reservation. We show that the tree routing property extends to the new problem, and give a 24.92-approximation algorithm for it. We also show that the undirected uncapacitated single-source minimum concave-cost (fractional) flow problem has always the tree property when the cost function has some property of symmetry.

1. INTRODUCTION

The symmetric Virtual Private Network Design (sVPN) problem is defined as follows. We are given an undirected network with costs on the edges, a set of terminals, and an upper bound for each terminal limiting the cumulative amount of traffic it can send or receive. The goal is to find a so-called virtual private network: a “subnetwork” of the given network, allowing the terminals to communicate with each other. The bounds implicitly describe the set of traffic demands that the virtual private network should support; such sets of traffic demands are called valid. The desired virtual private network consists of (i) a path for each unordered pair of terminals and (ii) a minimum cost reservation of capacities on the edges of the network so that all the valid sets of traffic demands can be routed along the selected paths. In the version of the problem which has been considered until now, the contribution of an edge to the total cost is proportional to the capacity reservation for that edge. In this paper, we propose a generalization of the problem where the cost function allows for economies of scales.

It was shown by Fingerhut, Suri and Turner [3] and later, independently, by Gupta, Kleinberg, Kumar, Rastogi and Yener [7] that sVPN can be solved in polynomial time if it has the tree routing property, that is, each instance has an optimal solution whose support is a tree. It was conjectured that sVPN has the tree routing property, see, e.g., Erlebach and Rüegg [2], Italiano, Leonardi and

Date: Tue Dec 2 16:00:00 CET 2008.
GO & LS: Dipartimento di Ing. dell’Impresa, Università di Roma “Tor Vergata”, Rome, Italy.
Emails: {oriolo, sanità}@disp.uniroma2.it.
SF & DOT: Département de Mathématique, Université Libre de Bruxelles, Brussels, Belgium.
Emails: {sfiorini, dtheis}@ulb.ac.be.
Oriolo [10] and Hurkens, Keijsper and Stougie [9]. The conjecture was recently solved affirmatively by Goyal, Olver and Shepherd [4] by settling an equivalent conjecture, due to Grandoni, Kaibel, Oriolo and Skutella [6], claiming that another problem called the Pyramidal Routing (PR) problem has the tree routing property. The Pyramidal Routing problem may be viewed as an undirected uncapacitated single-source minimum concave-cost flow problem (see e.g. [13]), where the cost of a flow $x$ is $\sum_{e \in E} c_e g(x_e)$, where $g$ is a pyramid-shaped function (see Fig. 1.(a)).

In this paper we investigate a natural generalization of sVPN where the cost per unit of capacity may decrease if a larger amount of capacity is reserved. More precisely, we define the concave symmetric Virtual Private Network Design (csVPN) problem as sVPN, but the contribution of each edge to the total cost is now proportional to some arbitrary fixed concave, non-decreasing function $f$ of the capacity reservation. For linear $f$ one recovers sVPN. However, the problem is APX-hard for general $f$ due to the fact that it contains the minimum Steiner tree problem as a special case (see Section 1.1 for details).

In order to study csVPN we introduce three problems that we respectively call the Concave Routing (CR) problem, the axis-symmetric Concave Routing (sCR) problem, and the non-decreasing Concave Routing (ndCR) problem, two of which are generalizations of PR. All these problems may also be viewed as undirected uncapacitated single-source minimum concave-cost flow problems: for CR, $g$ is just a non-negative concave function; for sCR, $g$ is concave and axis-symmetric (see Fig. 1(b)); and for ndCR, $g$ is concave and non-decreasing (for proper definitions, see Section 1.1). The various problems considered here and their relationships are are illustrated in Fig. 2.

Our main contributions are as follows. First, we show that csVPN has the tree routing property. Our proof goes as follows. First, we build upon the result by Goyal et al. [4] to show that sCR has the tree routing property. Then we show that there is an equivalence between csVPN and sCR, so that csVPN has the tree routing property too. Second, we study approximation algorithms for csVPN. For general $f$, using known results on the so-called Single Source Buy at Bulk (SSBB) problem by Grandoni and Italiano [12], we give a 24.92-approximation algorithm. For a restricted
class of functions $f$, by reducing to the so-called Single Source Rent or Buy (SSRB) problem, we show that a 2.92-approximation algorithm exists.

We also draw attention on the following fact. It is shown in [4] that there always exists an optimal solution to $\CR$ that is unsplittable, i.e., that routes all flows from the source to a terminal on a unique path, even when we allow fractional flows. A natural question then is whether the tree property holds in general for (fractional) $\CR$. Our results show that that is the case when $g$ is either non-decreasing or symmetric, while, in general, even if the input graph is a cycle and some conditions are imposed on $g$, other than being non-decreasing or axis-symmetric, the tree property does not hold.

1.1. Detailed description of the problems. In this paper, we mainly consider three routing problems: the symmetric Virtual Private Network Design ($s\VPN$) problem, its generalization with concave costs: the concave symmetric Virtual Private Network Design ($cs\VPN$) problem, and the Concave Routing ($CR$) problem.

We now describe the problems in detail. All the problems involve a (finite) simple, undirected, connected graph $G = (V, E)$ that represents a communication network. The graph comes with two vectors: a vector $c \in \mathbb{R}^E$ describing the edge costs and a vector $b \in \mathbb{Z}^V$ providing some information on the traffic that each vertex sends or receives (the exact interpretation depends on the problem). A vertex $v$ with $b_v > 0$ is referred to as a terminal. We denote the set of terminals by $W$. Also, we let $B$ be the sum of all components of $b$. In other words,

$$W := \{v \in V \mid b_v > 0\}, \text{ and}$$
$$B := \sum_{v \in V} b_v.$$

In the symmetric Virtual Private Network design ($s\VPN$) problem, the vertices of $G$ want to communicate with each other. However, the exact amount of traffic between pairs of vertices is not known in advance. Instead, for each vertex $v$ the cumulative amount of traffic that it can send or receive is bounded from above by $b_v$. The general aim is to install minimum cost capacities on the
edges of the graph supporting any possible communication scenario, where the cost for installing one unit of capacity on edge \( e \) equals its cost \( c_e \).

A set of traffic demands \( D = \{ d_{uv} \mid \{ u, v \} \subseteq W \} \) specifies for each unordered pair of terminals \( \{ u, v \} \subseteq W \) the amount \( d_{uv} \in \mathbb{R}_+ \) of traffic between \( u \) and \( v \). A set \( D \) is valid if it respects the upper bounds on the traffic of the terminals. That is,

\[
\sum_{u \in W} d_{uv} \leq b_v \quad \text{for all terminals } v \in W.
\]

A solution to the instance of \( s\text{VPN} \) defined by the triple \( (G, b, c) \) consists of a collection of paths \( P \) containing exactly one \( u-v \) path \( P_{uv} \) in \( G \) for each unordered pair \( u, v \) of terminals, and a vector \( \gamma \in \mathbb{R}_+^E \) describing the capacity to be installed on each edge. Such a set of paths \( P \), together with capacity reservations \( \gamma \), is called a virtual private network. A virtual private network is feasible if all valid sets of traffic demands can be routed without exceeding the reserved capacities, in case all traffic between terminals \( u \) and \( v \) is routed along path \( P_{uv} \), that is,

\[
\gamma_e \geq \sum_{\{u,v\} \subseteq W : e \in P_{uv}} d_{uv} \quad \text{for all edges } e \in E.
\]

Given a collection of paths \( P \) as above, one may compute in polynomial time the capacity reservations \( \gamma \) in order to obtain a feasible virtual private network by performing \( |E| \) fractional \( b \)-matching computations, see Gupta et al. [7] and Italiano et al. [10] for details.

The concave symmetric Virtual Private Network Design (\( cs\text{VPN} \)) problem is defined similarly as the \( s\text{VPN} \) problem. The total cost of virtual private network \((P, \gamma)\) is now

\[
\sum_{e \in E} c_e f(\gamma_e),
\]

where \( f \) is concave, non-decreasing and such that \( f(0) = 0 \). An instance of \( cs\text{VPN} \) is described by a quadruple \( (G, b, c, f) \). We assume we are given oracle access to the function \( f \). That is, we are given access to a subroutine that, given a rational \( x \in [0, B] \), returns a non-negative rational \( f(x) \) whose size is polynomial in the size of \( x \). The computation of \( f(x) \) is assumed to take constant time.

The \( cs\text{VPN} \) problem is APX-hard. In fact, the minimum Steiner tree problem is a restriction of \( cs\text{VPN} \): let \( b_v := 1 \) for each terminal and \( b_v := 0 \) otherwise, and then let \( f(0) := 0 \) and \( f(x) := 1 \) for \( x \in (0, B] \).

In the Concave Routing (\( \text{CR} \)) problem, one of the terminals is marked as a root. We denote the root by \( r \) (therefore, \( b_r > 0 \)). For each vertex \( v \), the number \( b_v \) describes the actual demand at the vertex.

A solution to the \( \text{CR} \) problem consists of a collection \( P \) of simple \( r-v \) paths \( P_v \), one path for each terminal \( v \) distinct from the root. We call such a collection a routing. We denote by \( x_e(P) \) the amount of flow routed on the edge \( e \) by \( P \), thus

\[
x_e(P) := \sum_{v \in W \setminus \{r\} : e \in P_v} b_v \quad \text{for all edges } e \in E.
\]
The cost of a routing is then:

\[(2) \sum_{e \in E} c_e g(x_e(P)),\]

where \(g: [0, B] \to \mathbb{R}_+\) is a concave function such that \(g(0) = 0\). An instance of CR is then defined by a quintuple \((G, r, b, c, g)\). (As for VPN, we assume we are given oracle access to \(g\).)

In particular, we are interested in the following restrictions of the CR problem. The instances of the non-decreasing Concave Routing (ndCR) problem are those for which \(g\) is non-decreasing. In this case, we use the letter \(f\) instead of \(g\) whenever possible. The instances of the axis-symmetric Concave Routing (sCR) problem are those for which \(g\) is (axis-)symmetric, that is, \(g(B-x) = g(x)\) for all \(x \in [0, B]\). In this case, we use the letter \(h\) instead of \(g\) whenever possible. Finally, the instances of the Pyramidal Routing (PR) problem \([6]\) are those for which \(g(x) = \min\{x, B-x\}\) for all \(x \in [0, B]\). In this case, we use the letter \(p\) instead of \(g\). Notice that PR is a special case of sCR. Notice also that both sCR and ndCR are NP-hard because they admit the minimum Steiner tree problem as a special case.

Finally, we also define the fractional version of CR (denoted by frac-CR), where we allow, for each terminal \(v \neq r\), to fractionally split the \(b_v\) units of flow from \(r\) to \(v\) along several \(r-v\) paths. Formally, a fractional routing \(\mathcal{P}\) specifies, for each terminal \(v \neq r\), a set \(\mathcal{P}_v\) of simple \(r-v\) paths and, for each path \(P \in \mathcal{P}_v\), an amount of flow \(\beta_v(P) \in \mathbb{R}_+\) such that \(b_v = \sum_{P \in \mathcal{P}_v} \beta_v(P)\). The cost of a routing is as in Eq. (2) above, with \(x_v(P) := \sum_{v \in \mathcal{W} \setminus \{r\}} \sum_{P \in \mathcal{P}_v, e \in P} \beta_v(P)\).

A feasible solution to one of the problems described above is a tree solution if the support of the capacity vector \(\gamma\) or the union of the paths in \(\mathcal{P}\) induces a tree in \(G\).

1.2. Previous work. Many fundamental results about VPN appear in Fingerhut et al. \([3]\) and Gupta et al. \([7]\). In particular, both papers show that computing a tree solution of minimum cost gives a 2-approximation algorithm for the problem. Such a solution can be obtained in polynomial time by solving a single all-pair shortest paths problem. It has been discussed \([8]\) and then conjectured in Erlebach et al. \([2]\) and in Italiano et al. \([10]\) that VPN has the tree routing property or, in other words, that the approximation algorithm mentioned above solves the problem exactly. This has become known as the VPN tree routing conjecture. The conjecture has first been proved for the case of cycles \([9,5]\), and then in general graphs \([4]\).

Goyal et al. \([4]\) prove the VPN tree routing conjecture by proving the PR has the tree routing property. This result was initially proposed as a conjecture by Grandoni et al. \([6]\), together with a proof that it implies the VPN tree routing conjecture. Remarkably, Goyal et al. \([4]\) also show that two results are equivalent, that is, VPN has the tree routing property if and only if PR has the tree routing property.

Our (fractional) ndCR is closely related to a known variant of the Single Source Buy at Bulk problem. See Section 3.2 for details.

1.3. Outline. In Section 2 we prove our main statements: csVPN and sCR have the tree routing property. The proof uses as a basis an equivalence, stated in Section 2.1 between csVPN and sCR. We show that, when \(b\) is a 0-1 vector, solving a csVPN instance \((G, b, c, f)\) amounts to solving an sCR instance of the form \((G, r, b, c, h)\) where \(r\) is one of the terminals and \(h\) is obtained by
symmetrizing $f$. Moreover, the $csVPN$ instance has an optimal solution that is a tree solution if and only if the corresponding $sCR$ instance has an optimal solution that is a tree solution. This allows us to focus only on $sCR$. By combining one decisive polyhedral observation with the fact that $PR$ has the tree routing property [4], we show that $sCR$ has the tree routing property, which then implies that $csVPN$ also has the tree routing property.

In Section 3 we give a constant factor approximation algorithm for $csVPN$. Our approximation algorithm works by reduction to the Single Source Buy at Bulk ($SSBB$) problem. The reduction is in two steps. We first observe in Section 3.2 that the approximation algorithm for $SSBB$ due to Grandoni and Italiano [12] gives an approximation algorithm for $ndCR$ with the same approximation factor. We then show in Section 3.3 how to turn any approximation algorithm for $ndCR$ into an approximation algorithm for $csVPN$ with the same approximation factor. Combining both steps, we obtain a $\rho$-approximation algorithm for $csVPN$ from the $\rho$-approximation algorithm for $SSBB$ due to Grandoni and Italiano [12], where $\rho = 24.92$. Using a subset of the tools developed, in Section 3.4 we give a 2.92-approximation algorithm for $csVPN$ when the function $f$ is of the type $f(x) = \min\{\mu x, M\}$ for positive constants $\mu$ and $M$. Here, we resort to the Single Source Rent or Buy ($SSRB$) problem, for which the best known approximation factor currently is 2.92 [14].

We conclude with some remarks on $CR$. Note that there always exists an optimal solution to $CR$ that is unsplittable [4], i.e., that routes all flows from the source to a terminal $v$ on a unique path $P_v$, even when we allow fractional flows. In Section 4 we show that, while the restriction of $CR$ to instances whose function $g$ is non-decreasing (resp., axis-symmetric) has the tree routing property, this is not the case in general, even if $G$ is a cycle and some extra conditions (other than being non-decreasing or axis-symmetric) are imposed on $g$.

2. The tree routing property

We show here that both $csVPN$ and $sCR$ have the tree routing property. We start by proving, in Section 2.1, an equivalence between the two problems when $b$ is a 0-1 vector. Then, in Section 2.2 we prove the tree routing property for $sCR$, and thus also for $csVPN$.

To make the terminology concise, we say that an instance of either $csVPN$ or $sCR$ has the tree routing property provided it admits an optimal solution with tree support.

2.1. Equivalence of $csVPN$ and $sCR$ instances in the binary case. We here restrict to instances where $b$ is a 0-1 vector. In this case, the number of terminals is $B$ and, for any routing $P$, there are precisely $x_e(P)$ paths in $P$ using the edge $e$. For $f: [0, B] \to \mathbb{R}_+$ concave and non-decreasing with $f(0) = 0$, we define

\[ h: [0, B] \to \mathbb{R}_+ : x \mapsto f(x) \quad \text{if } x \leq B/2, \]
\[ f(B - x) \quad \text{if } x > B/2. \]

Then $h$ is concave and axisymmetric and has $h(0) = 0$. The proof of the next lemma builds upon previous results of Gupta et al. [7], Grandoni et al. [6] and Goyal et al. [4].

**Lemma 1.** Let $(G, b, c, f)$ be a $csVPN$ instance with $b \in \{0, 1\}^V$, and $h$ as in (3). There exists a choice of a root $r \in W$ such that the $sCR$ instance $(G, r, b, c, h)$ has the same optimum value as
the csVPN instance. Moreover, for any such choice of root $r$, the corresponding sCR instance has the tree routing property if and only if the csVPN instance has the tree routing property.

Proof. Let $(P, \gamma)$ be a feasible virtual private network for $(G, b, c, f)$, with $P = \{P_{uv} \mid \{u, v\} \subseteq W\}$. For each possible root $r \in W$, let $P_r$ denote the routing consisting of all paths of $P$ one of whose ends is $r$. So $P_r := \{P_{rv} : v \in W \setminus \{r\}\}$. It is known [7] Theorem 3.2 [6] Lemma 3 that the following holds:

$$
\gamma_e \geq \frac{1}{B} \sum_{r \in W} \min\{x_e(P_r), B - x_e(P_r)\}.
$$

Since $f$ is concave and non-decreasing we have:

$$
\sum_{e \in E} c_e f(\gamma_e) \geq \sum_{e \in E} c_e \left( \frac{1}{B} \sum_{r \in W} \min\{x_e(P_r), B - x_e(P_r)\} \right) \\
\geq \frac{1}{B} \sum_{e \in E} c_e \sum_{r \in W} f(\min\{x_e(P_r), B - x_e(P_r)\}) = \frac{1}{B} \sum_{r \in W} \sum_{e \in E} c_e h(x_e(P_r)).
$$

Hence, the optimum value for the csVPN instance $(G, b, c, f)$ is at least the optimum value of the sCR instance $(G, r, b, c, h)$ for some choice of root $r \in W$.

Suppose conversely that we are given a routing $P_r$ for some sCR instance $(G, r, b, c, h)$, where this time $P_r := \{P_v \mid v \in W\}$. Following [4], we define a collection of paths $Q = \{Q_{uv} \mid \{u, v\} \subseteq W\}$, where $Q_{uv}$ is any $u$-$v$ path in the component of the symmetric difference $P_u \Delta P_v$ containing $u$ and $v$. Let $\delta_e$ be the minimum amount of capacity that we must install on each edge $e$ so that $(Q, \delta)$ is a feasible virtual private network for $(G, b, c, f)$. Goyal et al. [4] show that the following holds:

$$
\delta_e \leq \min\{x_e(P_r), B - x_e(P_r)\}.
$$

Since $f$ is non-decreasing, we have

$$
\sum_{e \in E} c_e f(\delta_e) \leq \sum_{e \in E} c_e f(\min\{x_e(P_r), B - x_e(P_r)\}) = \sum_{e \in E} c_e h(x_e(P_r)).
$$

Hence, the optimum value of the csVPN instance $(G, b, c, f)$ is at most the optimum value of any sCR instance of the form $(G, r, b, c, h)$. The statement easily follows. \qed

2.2. Proof of the tree routing property for CR. In this section, we will show how the tree routing property for sCR follows from the tree routing property for PR.

Theorem 2. The tree routing property holds for sCR.

Our approach is simple and geometric: We associate polyhedra with instances of sCR in such a way that the tree routing property for an instance can be expressed as a property of the extreme points of the associated polyhedron. We then show how the transition from the pyramidal function to an arbitrary concave axis-symmetric function $h$ amounts to a transformation of the corresponding polyhedra, which preserves the property of the extreme points.

Recall that, for a set $Z \subseteq \mathbb{R}^E$, the dominant $\text{dom} Z$ of $Z$ is defined as follows:

$$
\text{dom} Z := \{z' \in \mathbb{R}^E \mid \text{there exists some } z \in Z \text{ with } z' \geq z\}.
$$
Given, for every instance \( (\tilde{G}, \tilde{r}, \tilde{b}, \tilde{c}, h) \) such that \( \tilde{b} \) is not a 0-1 vector, we define a new instance \((\tilde{G}, \tilde{r}, \tilde{b}, \tilde{c}, h)\), as follows. For each terminal \( v \) with \( b_v \geq 2 \), we add \( k := b_v \) pendant edges \( vu_1, \ldots, vu_k \) with cost zero to the graph. Then, we let \( \tilde{b}_v := 0 \) and \( \tilde{b}_{u_i} := 1 \) for \( i = 1, \ldots, k \).
Finally, we let \( r \) be one of the new vertices pending from \( r \) except if \( b_r = 1 \) in which case we let \( \tilde{r} = r \). Since the new instance has an optimal solution that is a tree solution, it follows that also the original instance has an optimal solution that is a tree solution. \( \square \)

**Corollary 6.** The tree routing property holds for \( \text{csVPN} \).

**Proof.** First, consider an \( \text{csVPN} \) instance \((G, b, c, f)\) with \( b_v \in \{0, 1\} \) for each \( v \in V \). Here the statement follows from Lemma 1 and Theorem 2. The case where some terminals have demand greater than 1 can be reduced to the previous one by the same arguments as in the proof of Theorem 2. \( \square \)

### 3. Approximation Algorithms

#### 3.1. Preliminaries

As observed by Goyal et al. [4, Lemma 2.2], \( \text{frac-CR} \)—the fractional version of \( \text{CR} \)—always admits an optimal solution that is unsplittable (i.e., that routes all demands to a terminal \( v \) on a unique path \( P_v \)). Here we state and prove an algorithmic version of this result.

**Lemma 7.** Every instance of \( \text{frac-CR} \) has an unsplittable optimal solution. Moreover, given a fractional solution we can build an unsplittable one that does not cost more, in time polynomial in the size of the instance plus the size of the given fractional solution.

**Proof.** Let \( \tilde{P} \) be a fractional routing for some instance \((G, r, b, c, g)\) of \( \text{frac-CR} \). Consider some terminal \( v \). Let \( \{P_1, \ldots, P_t\} \) denote the set of all \( r-v \) paths used by \( \tilde{P} \), and let \( \beta_v(P_i) \) denote the flow on path \( P_i \), for \( i = 1, \ldots, t \). From \( \tilde{P} \), we define \( t \) routings \( \tilde{P}_1, \ldots, \tilde{P}_t \), as follows. For \( i \in \{1, \ldots, t\} \), the routing \( \tilde{P}_i \) routes all \( b_v \) units of demand to \( v \) on the single path \( P_i \) and otherwise behaves as \( \tilde{P} \). Thus, for each edge \( e \in E \), we have:

\[
x_e(\tilde{P}) = \sum_{i=1}^{t} \frac{\beta_v(P_i)}{b_v} x_e(\tilde{P}_i).
\]

By the concavity of the cost function \( x \mapsto \sum_{e \in E} c_e g(x_e) \), there exists \( i \in \{1, \ldots, t\} \) such that \( \tilde{P}_i \) does not cost more than \( \tilde{P} \). The result then follows by induction. \( \square \)

#### 3.2. An approximation algorithm for \( \text{ndCR} \)

Our approximation algorithm for \( \text{csVPN} \) is based on an approximation algorithm for \( \text{ndCR} \). Notice that the latter problem is also APX-hard. As we show below, there exists an approximation-factor preserving reduction from \( \text{ndCR} \) to the Single Source Buy at Bulk (SSBB) problem. The latter is defined as follows: we are given an undirected graph \( G = (V, E) \) with edge costs \( c \in \mathbb{R}^E_+ \), where each vertex \( v \in V \) wants to exchange an amount of flow \( b_v \in \mathbb{Z}_+ \) with a common source vertex \( r \). In order to support the traffic, we can install cables on edges. Specifically we can choose among \( k \) different cables: each cable \( i \in \{1, \ldots, k\} \) provides \( \mu(i) \) units of capacity at price \( p(i) > 0 \). For each \( i \in \{1, \ldots, k-1\} \), it is assumed that \( \mu(i) < \mu(i+1) \) and \( \frac{p(i)}{\mu(i)} \geq \frac{p(i+1)}{\mu(i+1)} \). The latter inequality is referred to as the economy of scale principle. An instance of \( \text{SSBB} \) is therefore defined by a quintuple \((G, r, b, c, K)\), where \( K = \{\mu(i), p(i)\mid i = 1, \ldots, k\} \) describes the different cable types.

A solution to \( \text{SSBB} \) consists of a multiset \( \kappa_e \) of cables to install on each edge \( e \in E \). Repetitions are allowed, that is, several cables of the same type can be installed on some edge.
We point out that there is some confusion in the literature in the definition of \( \text{SSBB} \), because in some papers \( \text{SSBB} \) is defined as above, and in some other papers the \( \text{SSBB} \) problem is defined as the problem we call \( \text{frac-ndCR} \). In this paper, when referring to \( \text{SSBB} \) we always mean the version with cables.

Here, we show how to adapt the 24.92-approximation algorithm which Grandoni and Italiano [12] have given for \( \text{SSBB} \), in order to obtain an algorithm with the same approximation ratio for \( \text{ndCR} \).

We start with a description of a simple approximation preserving reduction from \( \text{ndCR} \) to \( \text{SSBB} \). Let \( I = (G, r, b, c, f) \) be an instance of \( \text{ndCR} \). Consider the instance \( J = (G, r, b, c, K) \) of \( \text{SSBB} \) obtained by setting: \( K := \{(1, f(1)), (2, f(2)), \ldots, (B, f(B))\} \). The capacity of the cables are non-decreasing because \( f \) is non-decreasing. Since \( f(0) = 0 \) and \( f \) is concave, \( x \mapsto f(x)/x \) is non-increasing, and thus the economy of scale principle holds. We point out that the size of \( J \) is not always bounded by a polynomial in the size of \( I \), because we assume \( f \) is given via an oracle.

We address this issue below, after studying the approximation properties of the reduction.

**Lemma 8.** Let \( I \) and \( J \) be as described above. There is an algorithm that transforms a solution \( \kappa \) to \( J \) to a solution to \( I \), such that the ratio between the cost of \( \kappa \) and the optimum for \( J \) is at most the ratio between the solution to \( I \) and the optimum for \( I \). Moreover, this algorithm runs in time polynomial in the sum of the sizes of \( \kappa \) and \( I \). (Thus if the size of \( \kappa \) is polynomial in the size of \( I \), this algorithm is polynomial in the size of \( I \).)

**Proof.** We prove the statement by showing that: (i) Given a solution to \( I \) there exists a solution to \( J \) of the same cost (ii) from a solution \( \kappa \) to \( J \) one can build, in time polynomial in the sizes of \( I \) and \( \kappa \), a solution to \( I \) that does not cost more.

(i) Each solution \( \mathcal{P} \) to \( I \) yields a solution to \( J \) of the same cost: Install on each edge \( e \) a single cable of capacity \( x_e(\mathcal{P}) \in \mathbb{Z}_+ \).

(ii) Let \( \kappa \) be a solution to \( J \). Consider an edge \( e \) and let \( \gamma_e := \sum_{i \in \kappa_e} \mu(i) = \sum_{i \in \kappa_e} i \). Using flow techniques, we can compute, in time polynomial in the sizes of \( I \) and \( \kappa \), a fractional solution \( \mathcal{P} \) to \( I \) such that \( x_e(\mathcal{P}) \leq \gamma_e \) for all \( e \in E \). By applying Lemma 7 from \( \mathcal{P} \), we derive an unsplittable \( \text{ndCR} \) solution \( \mathcal{P} \) in polynomial time. This concludes the description of the algorithm.

From Lemma 7, we know that the cost of \( \mathcal{P} \) does not exceed that of \( \mathcal{P} \). We now prove that the cost of \( \mathcal{P} \) does not exceed that of \( \kappa \). For this, consider some edge \( e \). Without loss of generality, we may assume that \( \gamma_e \leq B \). Indeed, if this is not the case we can repeatedly replace some cable of capacity \( \mu(j) = j \) by a cable of capacity \( \mu(j - 1) = j - 1 \). This does not increase the cost of the solution.

Since \( x \mapsto f(x)/x \) is non-increasing, we have \( \frac{i}{\gamma_e} f(\gamma_e) \leq f(i) \) for each \( i \in \kappa_e \), thus

\[
\sum_{i \in \kappa_e} \frac{i}{\gamma_e} f(\gamma_e) \leq \sum_{i \in \kappa_e} f(i), \quad \text{that is,} \quad f(\gamma_e) \leq \sum_{i \in \kappa_e} f(i).
\]

On the other hand, we have \( f(x_e(\mathcal{P})) \leq f(\gamma_e) \) because \( f \) is non-decreasing and \( x_e(\mathcal{P}) \leq \gamma_e \). Hence, we have \( f(x_e(\mathcal{P})) \leq \sum_{i \in \kappa_e} f(i) \). Because this holds for all edges \( e \), the cost of \( \mathcal{P} \) does not exceed that of \( \kappa \). The result follows.

We now give the approximation result for \( \text{ndCR} \).
Theorem 9. There exists a 24.92-approximation algorithm for ndCR.

To prove the theorem, we rely on a key fact used in the analysis of Grandoni and Italiano [12], which we now describe. First of all, Grandoni and Italiano rely on approximating instances of the Steiner tree problem. We suppose an approximation algorithm for this problem with ratio $\rho_{st}$ is available.

Let $\alpha := 3.1207$ and $\beta := 2.4764$. Given any instance $J := (G, r, b, c, K)$ of SSBB, if a subset $\{i_1, \ldots, i_{k'}\} \subseteq \{1, \ldots, k\}$ of cables is available with the properties $i_1 = 1$, $i_k' = k$ and, for all $t \in \{1, \ldots, k' - 2\}$, cable $i_{t+1}$ is the smallest such that

\begin{align}
(4a) & \quad p(i_{t+1} + 1) \geq \frac{\alpha}{\mu(i_{t+1})} p(i_t) \\
(4b) & \quad \frac{p(i_{t+1})}{\mu(i_{t+1})} \leq \frac{1}{\beta} \frac{p(i_t)}{\mu(i_t)};
\end{align}

we define a new SSBB instance $J' := (G, r, b, c, K')$ where

\begin{align}
(5) & \quad K' := \{(\mu(i_1), p(i_1)), \ldots, (\mu(i_{k'}), p(i_{k'}))\}.
\end{align}

With these definitions, we can extract the following fact from Section 3.1 of [12].

Lemma 10 ([12]). There is an algorithm with the following properties: Given an instance $J'$ as described above, the algorithm finds a solution to $J'$, whose cost is at most

\begin{align}
\rho := 1 + \max\left\{ (4 + \rho_{st}) + \rho_{st} \frac{\alpha + 1}{\alpha - 1}, (4 + \rho_{st})\alpha + \rho_{st} \frac{\alpha + 1}{\alpha - 1}, (2 + 2/\beta + \rho_{st}) + \frac{2\rho_{st}}{\alpha - 1} + (4 + \rho_{st})\beta \right\}
\end{align}

times the optimum of $J$. Moreover the algorithm runs in time polynomial in the size of $J'$.

With $\rho_{st} := 1.55$ (see [15]), following Grandoni and Italiano [12], we obtain $\rho := 24.92$. Now let us prove Theorem 9.

Proof of Theorem 9. We give an algorithm which, given oracle access to $f$, produces an instance $J'$ of SSBB. The point here is to find a list of cables $K'$ as in (5) satisfying (4a) and (4b), with respect to the instance $J$ used in Lemma 8 in time polynomial in $\log B$. We then invoke Lemma 10 to obtain a solution $\kappa$ to the SSBB instance $J'$. Clearly, $\kappa$ is a solution to $J$. Since the size of $\kappa$ is polynomial in the size of $I$, that lemma gives the desired solution to ndCR.

To construct the list of cables $K'$, we let $i_1 := 1$. If $i_t$ has been found, we search for the $(t + 1)$th cable $i_{t+1}$ as follows.

Firstly, since $f$ is increasing, given $p(i_t)$, a binary search in $\{i_t + 1, \ldots, B\}$ finds the smallest value $i'$ satisfying (4a) with $i_{t+1}$ replaced by $i'$. If no such $i'$ satisfies (4a), we let $i_{t+1} := k$ and $k' := t + 1$. If $i'$ does exist, since $x \mapsto f(x)/x$ is non-increasing, the smallest possible value for $i_{t+1}$ satisfying (4b) in the range $\{i', \ldots, B\}$ can be found by binary search. Again, if no $i_{t+1}$ satisfies (4b), we let $i_{t+1} := k$ and $k' := t + 1$. 

Recalling that \( \mu(i_t) = i_t \), from (4a) and (4b) it follows: \( i_{t+1} \geq \beta \cdot i_t \cdot \frac{f(i_{t+1})}{f(i_t)} \geq \beta \cdot i_t \cdot \frac{i_{t+1} + 1}{i_{t+1}} \geq \frac{1}{2} \beta \cdot i_t \). Therefore the number of selected cables is \( O(\log_{\frac{1}{2}} B) \) and each cable can be found in time \( O(\log B) \). The result follows.

3.3. An approximation algorithm for csVPN. In order to state our approximation algorithm for csVPN we need two further results from the literature.

First, let \((G, b, c, f)\) be an instance of the csVPN problem and let \(W\) denote the set of all terminals. Consider a tree \(T\) spanning all the terminals. For each pair of terminals \(\{u, v\} \subseteq W\) there is a unique \(u-v\) path in \(T\). These paths form a collection of paths that we denote \(P_T\).

It is straightforward to compute the minimum amount of capacity \(\gamma_T^e\) we have to reserve on each edge \(e\) of \(T\) in order to obtain a feasible virtual private network from \(P_T\) [7,10]. Indeed, the removal of each edge \(e\) from \(T\) determines a partition of the set of terminals \(W\) into two of its subsets, say \(W_1(e)\) and \(W_2(e)\). Assuming \(\sum_{v \in W_1(e)} b_v \leq \sum_{v \in W_2(e)} b_v\), we have \(\gamma_T^e = \sum_{v \in W_1(e)} b_v\) for all edges \(e \in E(T)\).

We denote the cost of the feasible virtual private network \((P_T, \gamma_T)\) by \(z(P_T, \gamma_T)\).

For any choice of root \(r \in V(T)\), one can similarly derive from \(T\) a tree solution to the ndcCR instance \((G, r, b_r, c, f)\), where we let \(b_{r,v} := b_v\) for all vertices \(v \neq r\), and \(b_{r,r} := \max\{b_r, 1\}\). We denote the resulting routing by \(P_r^T\) and its cost by \(z(P_r^T)\).

By orienting each edge \(e\) of \(T\) towards \(W_1(e)\), we turn \(T\) into an arborescence. Letting \(r\) denote the root of this arborescence, we have \(\gamma_T^e = x_e(P_r^T)\) for all edges \(e\) of \(T\). The next lemma easily follows (see Gupta et al. [7] Lemma 2.1, Italiano et al. [10] Lemma 2.4).

**Lemma 11.** Let \(T, \gamma_T\) and \(P_r^T\) be as above. There exists a vertex \(r\) of \(T\) such that \(\gamma_T^e = x_e(P_r^T)\) for all edges \(e\) of \(T\). In particular, for that choice of \(r\), we have \(z(P_T, \gamma_T) = z(P_r^T)\).

Next, suppose that we are given a solution \(P_r\) to an instance \((G, r, b_r, c, f)\) of ndcCR. As observed by Goyal et al. [4], we can build a feasible solution \((Q, \delta)\) to the instance \((G, b, c, f)\) of csVPN as follows: for each pair of terminals \(u, v\), choose the path \(Q_{uv}\) to be any path in \(P_u \Delta P_v\) from \(u\) to \(v\), where \(P_u\) and \(P_v\) respectively denote the unique \(r-u\) and \(r-v\) paths in \(P_r\). Define \(Q\) as the collection formed by all the paths \(Q_{uv}\). As observed above, we may efficiently deduce from \(Q\) the minimum capacity reservation \(\delta\) such that \((Q, \delta)\) is a feasible virtual private network. Let \(z(Q, \delta)\) denote the cost of this virtual private network. We will need the next lemma. We omit its proof because it is not difficult (see Goyal et al. [4] for a stronger result):

**Lemma 12.** Let \(P_r, Q\) and \(\delta\) be as above. We have \(\delta_e \leq x_e(P_r)\) for all edges \(e\) of \(G\). Thus \(z(Q, \delta) \leq z(P_r)\).

We are now ready to describe our approximation algorithm for csVPN. The input to the algorithm is a csVPN instance \((G, b, c, f)\).

**Theorem 13.** Algorithm 1 is a \(\rho\)-approximation algorithm for csVPN.

**Proof.** Consider an optimal solution to an instance \((G, b, c, f)\) of the csVPN problem with cost \(\text{OPT}(G, b, c, f)\). From Theorem 6 we know that there exists a tree \(T\) such that \(\text{OPT}(G, b, c, f) = \text{OPT}(T, \gamma_T)\)
Algorithm 1 Approximation algorithm for csVPN

1. For each \( r \in V \), compute a \( \rho \)-approximate solution \( P_r \) to the ndCR instance \((G, r, b_r, c, f)\).
2. Let \( r^* \) be such that \( z(P_{r^*}) = \min_{r \in V} z(P_r) \).
3. From \( P_{r^*} \), build a solution \((Q, \delta)\) to the csVPN instance \((G, b, c, f)\) as for Lemma 12.
4. Output \((Q, \delta)\).

\( z(P^T_r, \gamma^T) \). By Lemma 11, \( \min_{r \in V(T)} z(P^T_r) \leq z(P^T_\gamma) \). Since \( P^T_\gamma \) is a solution to the ndCR instance \((G, r, b_r, c, f)\), it follows that \( \min_{r \in V(T)} z(P^T_r) \geq \min_{r \in V} \text{OPT}(G, r, b_r, c, f) = \text{OPT}(\tilde{G}, \tilde{r}, b_\tilde{r}, c, f) \), for some \( \tilde{r} \in V \). (Here, \( \text{OPT}(G, r, b_r, c, f) \) denotes the cost of an optimal solution to the ndCR instance \((G, r, b_r, c, f)\), for \( r \in V \).) By choice of \( r^* \), \( z(P_{r^*}) \leq \rho \text{OPT}(G, \tilde{r}, b_\tilde{r}, c, f) \). From Lemma 12, \( z(Q, \delta) \leq \rho \text{OPT}(G, b, c, f) \), as desired.

By Corollary 9, we conclude that there exists a \( \rho \)-approximation algorithm for the csVPN problem (recall that \( \rho = 24.92 \)).

3.4. A 2.92-approximation for a restricted class of functions. Notice that Algorithm 1 preserves the function \( f \) when the approximation algorithm for ndCR is invoked. In particular, if \( f \) belongs to a restricted class of functions where ndCR has a small approximation factor, our algorithm will have same factor on the corresponding instances. We now give an illustration that relies on the Single Source Rent or Buy (SSRB) problem.

An instance of SSRB is given by a tuple \((G, r, b, c, \mu, M)\) consisting of a graph \( G \), a sink vertex \( r \), and demands \( b_\nu \) on the remaining vertices, edge costs \( c \), and two positive numbers \( \mu, M \). The goal is to install capacities on the edges to allow for the simultaneous routing of the demands from the source. On each edge, capacity can either be rented, which amounts to a cost \( c_\nu \mu x_\nu \) for renting \( x_\nu \) units of capacity. Alternatively, infinite capacity can be bought, which amounts to a cost of \( c_\nu M \) on this edge, regardless of the amount of capacity needed on the edge.

Define now from the positive numbers \( \mu, M \) a function \( f \) by letting \( f(x) := \min\{\mu x, M\} \). It can be readily checked that, for this particular function \( f \), the ndCR instance constructed in Algorithm 1 from a csVPN instance is, except for decomposing into paths, just the corresponding instance of the ndCR problem. Hence, our results imply an approximation-preserving reduction from csVPN—restricted to instances such that \( f(x) := \min\{\mu x, M\} \) for some positive numbers \( \mu \) and \( M \)—to SSRB.

The best known approximation algorithm for SSRB known to us is the one by Gupta et al. [14], which has an approximation factor of 2.92, as was shown by Eisenbrand, Grandoni, Rothvoß, and Schäfer [1].

4. A REMARK ON NON-AXIS-SYMMETRIC CONCAVE FUNCTIONS

It is known (see, e.g., [11]) that the tree routing property is satisfied by every CR instance such that \( g \) is non-decreasing, and it follows from our results that this also holds when \( g \) is axisymmetric. A natural question arises: is the tree property satisfied by all CR instances?

The example below shows that this is not the case, even if \( g(x) \leq g(B - x) \), for each \( x \in [0, B/2] \), and \( G \) is a cycle.
Example 14. Consider an instance \((G, r, b, c, g)\) of the \(c\)R problem, where \(G = (V, E)\) is a cycle with vertex set \(V = \{0, 1, 2, 3, 4\}\) and edge set \(E = \{\{i, i+1\} \mid i \in V\}\) (the sum is modulo 5). Let \(r := 0\); let \(b_i := 1\) for \(i \in V\); let \(c_e := M\) for \(e = \{3, 4\}\), \(c_e := M + \epsilon\) for \(e = \{0, 1\}\), \(c_e := 0\) otherwise. Finally, let \(g\) be defined as the linear interpolation of the following points: \(g(0) = 0, g(2) = 2, g(3) = 2 + 2\epsilon, g(5) = 0\). It is easy to check that \(g\) is concave, non-negative, non-axisymmetric and \(g(x) \leq g(B - x)\), for each \(x \in [0, B/2]\).

Consider the routing \(\mathcal{P}\) where the paths from 0 to \(i\) go counterclockwise (that is, have the edge \(\{0, 4\}\) as their first edge) for \(i = 1, 2, 3\), while the path from 0 to 4 goes clockwise. The cost of this solution is \((2 + \epsilon)M + \epsilon\), and it is easy to check that taking \(\epsilon\) and \(M\) respectively small and big enough, every tree routing costs more.

5. Conclusion

We would like to conclude with a problem. Let \(\mathcal{F}\) be an arbitrary set of concave non-decreasing functions \(\tilde{f} : [0, 1] \to \mathbb{R}_+\) with \(\tilde{f}(0) = 0\) (such that the coding size of \(\tilde{f}(x)\) is bounded by a polynomial in the coding size of \(x\) for every rational \(x\)). We ask the question whether \(cs\)VPN is NP-hard if the oracle function \(f\) is required to be of the form \(f(x) = \tilde{f}(x/B)\), for an \(\tilde{f} \in \mathcal{F}\). The following is known:

- If \(1_{[0,1]} \in \mathcal{F}\), then \(cs\)VPN is NP-hard (here \(1_{[0,1]}\) denotes the function which is 0 in 0 and 1 otherwise).
- If \(\mathcal{F}\) includes functions of the form \(x \mapsto \min\{\mu x, 1\}\) for unbounded values of \(\mu \geq 0\), then \(cs\)VPN is NP-hard.
- If \(\mathcal{F}\) includes all strictly increasing piece-wise affine functions, then \(cs\)VPN is NP-hard.

While the first two include the Steiner tree problem as a special case (i.e., a particular choice of function \(\tilde{f}\), depending on \(B\)), the third is a consequence of the APX-hardness of the Steiner tree problem by approximating the function \(1_{[0,1]}\) by functions in \(\mathcal{F}\). The question we pose is the following: Is \(cs\)VPN NP-hard for other sets \(\mathcal{F}\), and in particular when \(\mathcal{F}\) contains all \(q\)-Lipschitz continuous functions, where \(q\) is a fixed constant.

Note added in preparation: Following the results in this manuscript, an alternative proof of the fact that the tree property holds for \(cs\)VPN has been given [5]. This proof, however, does not show that it also holds also for \(s\)CR.

References


