SIAM J. DISCRETE MATH.
Vol. 24, No. 3, pp. 1080–1090

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*, ACM, New York, 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. Algorithms*, 24 (1997), pp. 287–309] and Gupta et al. [*Proc. STOC*, ACM, New York, 2001], `sVPN` can be solved in polynomial time. In this paper we investigate an APX-hard generalization of `sVPN`, where the contribution of each edge to the total cost is proportional to some non-negative, concave, and nondecreasing function of the capacity reservation. We show that the tree routing property extends to the new problem and give a constant-factor approximation algorithm for it. We also show that the undirected uncapacitated single-source minimum concave-cost flow problem has the tree routing property when the cost function has some property of symmetry.

Key words. network design, routing problem, approximation algorithm, concave cost

AMS subject classifications. 90C27, 68M10, 68W25

DOI. 10.1137/090749700

1. Introduction.

All the problems considered in this paper involve a (finite) simple, undirected, connected graph $G = (V, E)$ that represents a communication network. The graph comes with a vector $c \in \mathbb{Q}_+^E$ describing per-unit edge costs and a vector $b \in \mathbb{Z}_+^V$ pertaining to the traffic departing from or arriving at each vertex; 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. Thus, $W = \{v \in V \mid b_v > 0\}$ and $B = \sum_{v \in V} b_v$.

In the *symmetric virtual private network design* (`sVPN`) problem, the vertices 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 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 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 $d_{uv} \in \mathbb{Q}_+$ of traffic between u and v. A set

*Received by the editors February 13, 2009; accepted for publication (in revised form) June 16, 2010; published electronically September 7, 2010. This work was partially supported by the Actions de Recherche Concertées (ARC) fund of the *Communauté française de Belgique*, the *Fonds National de la Recherche Scientifique* (F.R.S.FNRS), and the project *Robust Network Design* of the Swiss National Science Foundation.

http://www.siam.org/journals/sidma/24-3/74970.html

†Département de Mathématique, Université Libre de Bruxelles, Brussels, Belgium (sfiorini@ulb.ac.be).

‡Dipartimento di Ing. dell'Impresa, Università di Roma "Tor Vergata," Rome, Italy (oriolo@disp.uniroma2.it).

§Dipartimento di Ing. dell'Impresa, Università di Roma "Tor Vergata," Rome, Italy. New address: Institute of Mathematics, EPFL, Lausanne, Switzerland (laura.sanita@epfl.ch).

¶Département de Mathématique, Université Libre de Bruxelles, Brussels, Belgium. New address: Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg, Germany (dirk.theis@ovgu.de).

D is *valid* if it respects the upper bounds on the traffic of the terminals. That is,

$$\sum_{u \in W} d_{uv} \leq b_v \qquad \text{for all terminals } v \in W.$$

A solution to the instance of sVPN defined by the triple (G, b, c) consists of a collection of paths $\mathcal{P}$ containing exactly one u–v path P_{uv} in G for each unordered pair u, v of terminals, and a vector $\gamma \in \mathbb{Q}_+^E$ describing the capacity to be installed on each edge. Such three paths $\mathcal{P}$, together with capacity reservations γ, 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,

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

Given a collection of paths $\mathcal{P}$ as above, one may compute in polynomial time the minimum capacity reservations γ_e for $e \in E$ in order to obtain a feasible virtual private network [8, 12].

The *concave symmetric virtual private network design* (csVPN) problem is defined similarly as sVPN. The total cost of virtual private network $(\mathcal{P},\gamma)$ is now

$$\sum_{e \in E} c_e \, f(\gamma_e), \tag{1.1}$$

where $f\colon [0, B] \to \mathbb{R}_+$ is concave, nondecreasing, and such that $f(0) = 0$. (We assume we are given oracle access to f; see section 1.4 below.) An instance of csVPN is described by a quadruple (G, b, c, f).

In the *concave routing* (CR) problem, one of the terminals is marked as *root*. We denote the root by r. For each vertex v, the number b_v describes the *demand* at the vertex. We remark that, by the choice of r, there is a demand $b_r > 0$ at the root. This is a dummy demand that does not play any role in the problem.[1]

A solution to CR consists of a collection $\mathcal{P}$ 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 $x_e(\mathcal{P})$ the amount of flow routed on the edge e by $\mathcal{P}$. Thus, $x_e(\mathcal{P}) = \sum_{v \in W \setminus \{r\} : e \in P_v} b_v$. The cost of a routing is then

$$\sum_{e \in E} c_e \, g(x_e(\mathcal{P})), \tag{1.2}$$

where $g\colon [0, B] \to \mathbb{R}_+$ is a concave function such that $g(0) = 0$. (Once again, we assume that we are given oracle access to g.) An instance of CR is then defined by a quintuple (G, r, b, c, g). We remark that CR may be viewed as an undirected uncapacitated single-source minimum concave-cost flow problem [10].

We are interested in the following restrictions of CR. The instances of the *nondecreasing concave routing* (ndCR) problem are those for which g is nondecreasing. 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

[1] We use this convention in order to be consistent with previous published work [7].

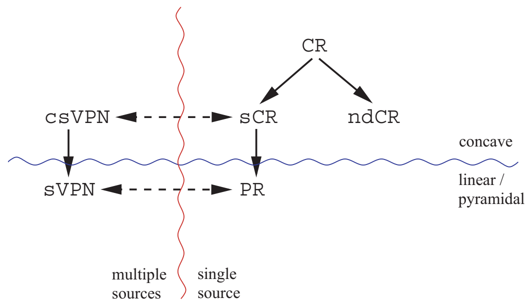

FIG. 1.1. *The problems considered in this article. Bold arrows indicate specialization, and dashed arrows indicate "equivalence."*

h instead of g whenever possible. Finally, the instances of the *pyramidal routing* (`PR`) problem [7] 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.

The various problems considered here and their relationships are illustrated in Figure 1.1. Notice that `csVPN`, `sCR`, and `ndCR` are all APX-hard because they admit the minimum Steiner tree problem as a special case.

A feasible solution to one of the problems described above is a *tree solution* if the support of the capacity vector γ or the union of the paths in $\mathcal{P}$ induces a tree in G. To make the terminology concise, we say that an instance of either `csVPN` or `sCR` has the tree routing property provided one of its optimal solutions is a tree solution.

1.1. Previous work. It was shown by Fingerhut, Suri, and Turner [3] and later, independently, by Gupta et al. [8] that `sVPN` can be solved in polynomial time if it has the tree routing property; that is, each instance has an optimal solution that is a tree solution.[2] Subsequently, it has been discussed [9] and then conjectured [2, 12] that `sVPN` has the tree routing property. This has become known as the *VPN tree routing conjecture*. The conjecture has first been proved for the case of cycles [11, 7], and then in general graphs [4].

Goyal, Olver, and Shepherd [4] prove the VPN tree routing conjecture by proving that `PR` has the tree routing property. This result was initially proposed as a conjecture by Grandoni et al. [7], together with a proof that it implies the VPN tree routing conjecture. Remarkably, Goyal, Olver, and Shepherd [4] also show that two results are equivalent; that is, `sVPN` has the tree routing property if and only if `PR` has the tree routing property.

1.2. Our contribution. First, we show that `csVPN` has the tree routing property. Our proof goes as follows. On the one hand, we build upon the result by Goyal, Olver, and Shepherd [4] to show that `sCR` has the tree routing property. On the other hand, we show that `sCR` has the tree routing property if and only if `csVPN` has the tree routing property.

Second, we study approximation algorithms for `csVPN`. For general f, using known results on the so-called single source buy at bulk (`SSBB`) problem [14, 6], 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 [1], we show that a 2.92-approximation algorithm exists.

[2]Such a solution can be obtained in polynomial time by solving a single all-pair shortest paths problem.

Third, although sCR and ndCR both have the tree routing property, we show that this is not the case for the general CR problem.

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 an 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 SSBB problem. The reduction is in two steps. First, we observe in section 3.1 that the approximation algorithm for SSBB due to Grandoni and Italiano [6], which is a variation of the algorithm of Gupta et al. [14], gives an approximation algorithm for ndCR with the same approximation factor. (It directly follows from known results from the literature that ndCR has a constant-factor approximation algorithm. The aim of section 3.1 is merely to halve the resulting approximation factor.) Then, we show in section 3.2 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 ρ-approximation algorithm for csVPN from the ρ-approximation algorithm for SSBB [6], where $\rho = 24.92$. Using a subset of the tools developed, we give a 2.92-approximation algorithm for csVPN when the function f is to be of the type $f(x) = \min\{\mu x, M\}$ for positive constants μ and M. Here, we resort to the SSRB problem, for which the best known approximation factor currently is 2.92 [14, 1].

In section 4, we give an instance of CR such that no tree solution is optimal, thereby showing that CR does not have the tree routing property.

1.4. Fractional problems and value-giving oracles. Before starting section 2, we conclude this section by providing necessary extra details.

We 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 (1.2) above, with $x_e(\mathcal{P}) := \sum_{v \in W \setminus \{r\}} \sum_{P \in \mathcal{P}_v : e \in P} \beta_v(P)$.

It results from the concavity of g (see, e.g., Goyal, Olver, and Shepherd [4, Lemma 2.2]) 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. Therefore, the frac-CR problem and CR problem are essentially equivalent.

The problem frac-ndCR is defined similarly. This last problem is closely related to a known variant of the SSBB problem; see section 3.1 for details.

Finally, in the csVPN (resp., CR) problem, we assume that we are given oracle access to the function f (resp., g). That is, we are given access to a subroutine that, given a rational $x \in [0, B]$, returns a non-negative rational $f(x)$ (resp., $g(x)$) whose size is polynomial in the size of x. The computation is assumed to take constant time.

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, that the tree routing property holds for csVPN if and only if it holds for sCR, provided that b is a 0-1 vector. Then, in section 2.2, we prove that the tree routing property holds for sCR, and thus also for csVPN, for any vector $b \in \mathbb{Z}_+^V$.

2.1. The tree routing property in the binary case. Here we restrict ourselves to instances where b is a 0-1 vector. In this case, the number of terminals is B and, for any routing $\mathcal{P}$, there are precisely $x_e(\mathcal{P})$ paths in $\mathcal{P}$ using the edge e. For $f\colon [0, B] \to \mathbb{R}_+$ concave and nondecreasing with $f(0) = 0$, we define

$$h\colon [0,B] \to \mathbb{R}_+\colon x \mapsto \begin{cases} f(x) & \text{if } x \le B/2, \\ f(B-x) & \text{if } x > B/2. \end{cases} \tag{2.1}$$

Then h is concave and axis symmetric and has $h(0) = 0$. The proof of the next lemma builds upon previous results of Gupta et al. [8], Grandoni et al. [7], and Goyal, Olver, and Shepherd [4].

LEMMA 2.1. *Let (G, b, c, f) be a* csVPN *instance with $b \in \{0,1\}^V$, and h as in* (2.1). *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, the corresponding* sCR *instance has the tree routing property if and only if the* csVPN *instance has the tree routing property.*

Proof. Let $(\mathcal{P}, \gamma)$ be a feasible virtual private network for (G, b, c, f), with $\mathcal{P} = \{P_{uv} \mid \{u,v\} \subseteq W\}$. For each possible root $r \in W$, let $\mathcal{P}_r$ denote the routing consisting of all paths of $\mathcal{P}$, one of whose ends is r. So $\mathcal{P}_r := \{P_{rv} : v \in W \setminus \{r\}\}$. It is known [8, Theorem 3.2], [7, Lemma 3] that the following holds:

$$\gamma_e \ge \frac{1}{B} \sum_{r \in W} \min\{x_e(\mathcal{P}_r), B - x_e(\mathcal{P}_r)\}.$$

Since f is concave and nondecreasing we have

$$\begin{aligned} \sum_{e \in E} c_e\, f(\gamma_e) &\ge \sum_{e\in E} c_e\, f\left(\frac{1}{B} \sum_{r\in W} \min\{x_e(\mathcal{P}_r), B - x_e(\mathcal{P}_r)\}\right) \\ &\ge \frac{1}{B} \sum_{e\in E} c_e \sum_{r \in W} f(\min\{x_e(\mathcal{P}_r), B - x_e(\mathcal{P}_r)\}) = \frac{1}{B} \sum_{r\in W} \sum_{e \in E} c_e\, h(x_e(\mathcal{P}_r)). \end{aligned}$$

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 $r \in W$. Note that, if $(\mathcal{P}, \gamma)$ is a tree solution, then $\mathcal{P}_r$ is also a tree solution for any $r \in W$. It is not difficult to see that, in this case, the cost of the routing $\mathcal{P}_r$ is not dependent on the root r. It follows that, given a tree solution to the csVPN instance (G, b, c, f), we can construct a tree solution to the sCR instance (G, r, b, c, h) that is not more costly, for any choice of root r.

Conversely, take any $r \in W$ and suppose that we are given a routing $\mathcal{P}_r$ for some sCR instance (G, r, b, c, h), where this time $\mathcal{P}_r := \{P_v \mid v \in W \setminus \{r\}\}$. Following [4], we define a collection of paths $\mathcal{Q} = \{Q_{uv} \mid \{u, v\} \subseteq W\}$, 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 δ_e be the minimum amount of capacity that we must install on each edge e so that $(\mathcal{Q}, \delta)$ is a feasible virtual private network for (G, b, c, f). Goyal, Olver, and Shepherd [4] show that the following holds:

$$\delta_e \leq \min\{x_e(\mathcal{P}_r), B - x_e(\mathcal{P}_r)\}.$$

Since f is nondecreasing, we have

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

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) for $r \in W$. Again, note that if $\mathcal{P}_r$ is a tree solution to (G, r, b, c, h), then $(\mathcal{Q}, \delta)$ is a tree solution to the csVPN instance (G, b, c, f). Therefore, given a tree solution to the sCR instance (G, r, b, c, h), we can construct a tree solution to the csVPN instance (G, b, c, f) that is not more costly. The statement easily follows. $\square$

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

THEOREM 2.2. *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 $\operatorname{dom} Z$ of Z is defined as follows:

$$\operatorname{dom} Z := \left\{ z' \in \mathbb{R}^E \mid \text{there exists some } z \in Z \text{ with } z \leq z' \right\}.$$

Here, and below, comparisons between vectors are componentwise. Given G, r, b, and h as above in the definition of sCR, a routing $\mathcal{P}$ defines a point $y(h, \mathcal{P}) \in \mathbb{R}^E_+$ by $y_e(h, \mathcal{P}) := h(x_e(\mathcal{P}))$ for all $e \in E$. We define the sCR *polyhedron* $P_{(G,r,b,h)}$ as the dominant of the convex hull of the points $y(h, \mathcal{P})$, where $\mathcal{P}$ ranges over all routings. Now, finding a routing that is minimum w.r.t. some non-negative cost vector c is equivalent to minimizing the linear function $y \mapsto c^T y$ over the sCR polyhedron. We note an easy consequence of this fact.

LEMMA 2.3. *Given G, r, b, and h, as above the following are equivalent:*

(i) *For every extreme point y of $P_{(G,r,b,h)}$, there exists a tree solution $\mathcal{T}$ such that $y = y(h, \mathcal{T})$.*

(ii) *For every $c \geq 0$, the* sCR *instance (G, r, b, c, h) has the tree routing property.*

We say that a mapping $\Phi\colon \mathbb{R}^E_+ \to \mathbb{R}^E_+$ is *concave* if $\Phi(tx + (1-t)y) \geq t\,\Phi(x) + (1-t)\,\Phi(y)$ holds for every $t \in [0,1]$ and $x, y \in \mathbb{R}^E_+$. Similarly, we say that such a mapping is *nondecreasing* if $x \leq y$ implies $\Phi(x) \leq \Phi(y)$. The key observation to realizing that the tree routing property for sCR is a consequence of the tree routing property for PR is the following.

LEMMA 2.4. *Let p denote the pyramidal function and h be as above. There exists a nondecreasing concave function $\Phi\colon \mathbb{R}^E_+ \to \mathbb{R}^E_+$ such that $\Phi(y(p, \mathcal{P})) = y(h, \mathcal{P})$ for all routings $\mathcal{P}$.*

Proof. For every e, we define $\Phi_e(y) := h(y_e)$ whenever $y_e \leq B/2$ and $\Phi_e(y) := h(B/2)$ if $y_e \geq B/2$. The properties are readily verified, since any axis-symmetric concave function $h\colon[0,B] \to \mathbb{R}_+$ is nondecreasing in the interval $[0, B/2]$, and $y_e(p, \mathcal{P})$ is always at most $B/2$. ◻

The final ingredient is the following elementary geometric fact.

LEMMA 2.5. *If $\Phi\colon \mathbb{R}_+^E \to \mathbb{R}_+^E$ is nondecreasing and concave, and Y is a finite set of points in $\mathbb{R}_+^E$, then every extreme point of $\operatorname{dom}\operatorname{conv}\Phi(Y)$ is the image under Φ of an extreme point of $\operatorname{dom}\operatorname{conv} Y$. In other words, Φ maps a subset of the extreme points of $\operatorname{dom}\operatorname{conv} Y$ onto the extreme points of $\operatorname{dom}\operatorname{conv}\Phi(Y)$.*

Proof. Consider an extreme point z of $\operatorname{dom}\operatorname{conv}\Phi(Y)$. If some point in $\Phi^{-1}(z)$ is an extreme point of $\operatorname{dom}\operatorname{conv} Y$, then we are done. Otherwise, pick any point y in $Y \cap \Phi^{-1}(z)$. By assumption, there exist extreme points $y_1, \dots, y_n \in Y \setminus \Phi^{-1}(z)$ and coefficients $\lambda_1, \dots, \lambda_n \geq 0$ with $\sum \lambda_j = 1$ such that $y \geq \sum_{j=1}^n \lambda_j y_j$. Hence, the assumptions on Φ imply $z = \Phi(y) \geq \Phi(\sum_{j=1}^n \lambda_j y_j) \geq \sum_{j=1}^n \lambda_j \Phi(y_j)$. Because $\Phi(y_j) \neq z$ for all j, the point z is not an extreme point of $\operatorname{dom}\operatorname{conv}\Phi(Y)$, a contradiction. ◻

Combining the previous two lemmas and this fact we obtain our theorem.

Proof of Theorem 2.2. We give the proof for 0-1 demands first. For this situation, Goyal, Olver, and Shepherd [4] have proven the tree routing property for all instances of `PR`. Lemma 2.3 implies that for every extreme point of $P_{(G,r,b,p)}$ there exists a tree solution defining it. By Lemmas 2.4 and 2.5, we know that this is also true for the extreme points of $P_{(G,r,b,h)}$. Another application of Lemma 2.3 yields the result for 0-1 demands.

Now consider an `sCR` instance (G, r, b, c, h) such that 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 $vu_1, \dots, vu_k$ with cost zero to the graph. Then, we let $\tilde{b}_v := 0$ and $\tilde{b}_{u_i} := 1$ for $i = 1, \dots, k$. Finally, we let $\tilde{r}$ 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. ◻

COROLLARY 2.6. *The tree routing property holds for* `csVPN`.

Proof. First, consider a `csVPN` instance (G, b, c, f) with $b_v \in \{0, 1\}$ for each $v \in V$. Here the statement follows from Lemma 2.1 and Theorem 2.2. 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 2.2. ◻

Remark. As pointed out by an anonymous referee, the results of this section still hold in case a concave function f_e (resp., h_e) is associated with each edge e of the graph, i.e., allowing different edges to have different functions.

3. Approximation algorithms.

3.1. An approximation algorithm for `ndCR`. Our approximation algorithm for `csVPN` is based on an approximation algorithm for `ndCR`. The approximation algorithm for `ndCR` is, in its turn, related to an approximation algorithm for the `SSBB` problem.

The latter problem is defined as follows: we are given a (finite, simple, undirected, connected) graph $G = (V, E)$ with edge costs $c \in \mathbb{Q}_+^E$, 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 $i \in \{1, \dots, k\}$ provides $\mu(i) \in \mathbb{Q}_+ \setminus \{0\}$ units of capacity at price $p(i) \in \mathbb{Q}_+ \setminus \{0\}$. For each $i \in \{1, \dots, k-1\}$, 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 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 SSBB consists of a multiset κ_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 SSBB, because in some papers SSBB is defined as above, and in some other papers the SSBB problem is defined as the problem we call frac-ndCR. In this paper, when referring to SSBB we always mean the version with cables. It is a known fact (see, e.g., Gupta *et al.* [14]) that from an approximation viewpoint, the two formulations are equivalent up to a factor of 2. However, we here show how to adapt the 24.92-approximation algorithm for SSBB described in [6], in order to obtain an algorithm with *the same* approximation ratio for ndCR.

THEOREM 3.1. *There exists a* 24.92*-approximation algorithm for* ndCR.

Proof. We start with a description of a simple approximation preserving reduction from ndCR to SSBB. Let $I = (G, r, b, c, f)$ be an instance of ndCR. Consider the instance $J = (G, r, b, c, K)$ of SSBB obtained by setting $K := \{(1, f(1)), (2, f(2)), \ldots, (B, f(B))\}$. The capacity of the cables are nondecreasing because f is nondecreasing. Since $f(0) = 0$ and f is concave, $x \mapsto f(x)/x$ is nonincreasing, and thus the economy of scale principle holds. It is easy to see that (i) given a solution to I there exists a solution to J of the same cost; (ii) from a solution κ to J one can build, in time polynomial in the sizes of I and κ, a solution to I that does not cost more. In other words, we could run the 24.92-approximation algorithm for SSBB on J and obtain a 24.92-approximate solution to I.

However, we point out that the size of J is not always bounded by a polynomial in the size of I (defined as the size of the quadruple (G, r, b, c); the function f is not taken into account when the size of I is computed), because B could be exponentially large. To address this issue, we rely on a key fact used in the analysis of Grandoni and Italiano [6], which we now describe. Given any instance $(\tilde{G}, \tilde{r}, \tilde{b}, \tilde{c}, \tilde{K})$ of SSBB, they select a subset $\{i_1, \ldots, i_{k'}\} \subseteq \{1, \ldots, k\}$ of cables with the following 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

$$p(i_{t+1} + 1) \geq \alpha\, p(i_t), \tag{3.1}$$

$$\frac{p(i_{t+1})}{\mu(i_{t+1})} \leq \frac{1}{\beta}\, \frac{p(i_t)}{\mu(i_t)}, \tag{3.2}$$

with $\alpha := 3.1207$ and $\beta := 2.4764$. Then, they find a 24.92-approximate solution to the SSBB instance using only cables in the following subset:

$$\tilde{K}' := \{(\mu(i_1), p(i_1)), \ldots, (\mu(i_{k'}), p(i_{k'}))\}, \tag{3.3}$$

with a running time polynomial in the size of $(\tilde{G}, \tilde{r}, \tilde{b}, \tilde{c}, \tilde{K}')$.

For our purpose, the point is therefore to find a list of cables K' as in (3.3) satisfying (3.1) and (3.2), with respect to the instance J, in time polynomial in $\log B$. 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.

First, since f is increasing, given $p(i_t)$, a binary search in $\{i_t + 1, \ldots, B\}$ finds the smallest value i' satisfying (3.1) with i_{t+1} replaced by i'. If no such i' satisfies (3.1), we let $i_{t+1} := k$ and $k' := t + 1$. If i' does exist, since $x \mapsto f(x)/x$ is nonincreasing, the smallest possible value for i_{t+1} satisfying (3.2) in the range $\{i', \ldots, B\}$ can be found by binary search. Again, if no i_{t+1} satisfies (3.2), we let $i_{t+1} := k$ and $k' := t + 1$.

Recalling that $\mu(i_t) = i_t$, from (3.1) and (3.2) it follows: $i_{t+1} \geq \beta \cdot i_t \cdot \frac{f(i_{t+1})}{f(i_t)} \geq \beta \cdot i_t \cdot \frac{i_{t+1}}{i_{t+1}+1} \frac{f(i_{t+1}+1)}{f(i_t)} \geq \frac{1}{2}\alpha \cdot \beta \cdot i_t$. Therefore the number of selected cables is $O(\log_{\frac{\alpha\beta}{2}} B) = O(\log B)$ and each cable can be found in time $O(\log B)$. The result follows. ∎

3.2. 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. Consider a tree T spanning all the terminals in W. 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 $\mathcal{P}^T$. It is straightforward to compute the minimum amount of capacity γ_e^T we have to reserve on each edge e of T in order to obtain a feasible virtual private network from $\mathcal{P}^T$. We denote $z(\mathcal{P}^T, \gamma^T)$ the cost of this virtual private network.

For any choice of root $r \in V(T)$, one can similarly derive from T a tree solution to the ndCR instance (G, r, b^r, c, f), where we let $b_v^r := b_v$ for all vertices $v \neq r$, and $b_r^r := \max\{b_r, 1\}$ (recall that in the definition of CR, we assume to have a positive, dummy demand at the root). We denote the resulting routing by $\mathcal{P}_r^T$ and its cost by $z(\mathcal{P}_r^T)$. The next lemma is known [8, Lemma 2.1], [12, Lemma 2.4]. For the sake of completeness, we give a sketch of its proof.

LEMMA 3.2. *Let T, $\mathcal{P}^T$, γ^T, and $\mathcal{P}_r^T$ (for $r \in V(T)$) be as above. Then, there exists a vertex r of T such that $\gamma_e^T = x_e(\mathcal{P}_r^T)$ for all edges e of T. For that choice of r, we have $z(\mathcal{P}^T, \gamma^T) = z(\mathcal{P}_r^T)$.*

Proof. Consider an edge e of T. The removal of e from T determines a partition of the set of terminals W into two of its subsets, say, $W_1(e)$ and $W_2(e)$. For definiteness, we assume that $W_1(e)$ and $W_2(e)$ are chosen in such a way that $\sum_{v \in W_1(e)} b_v \leq \sum_{v \in W_2(e)} b_v$. Then, the minimum capacity reservation γ_e^T for edge e is simply $\sum_{v \in W_1(e)} b_v$. By breaking ties consistently and orienting each edge $e \in E(T)$ towards $W_1(e)$, we can turn T into an arborescence. Letting r denote the root of this arborescence, we have $\gamma_e^T = x_e(\mathcal{P}_r^T)$ for all edges e of T. ∎

Second, suppose that we are given a solution $\mathcal{P}_r$ to an instance (G, r, b^r, c, f) of ndCR. As observed by Goyal, Olver, and Shepherd [4] and used in Lemma 2.1 above, we can build a feasible solution $(\mathcal{Q}, \delta)$ 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 $\mathcal{P}_r$. Define $\mathcal{Q}$ as the collection formed by all the paths Q_{uv}. As mentioned in the Introduction, we may efficiently deduce from $\mathcal{Q}$ the minimum capacity reservation δ such that $(\mathcal{Q}, \delta)$ is a feasible virtual private network. Let $z(\mathcal{Q}, \delta)$ 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, Olver, and Shepherd [4] for a stronger result).

LEMMA 3.3. *Let $\mathcal{P}_r$, $\mathcal{Q}$, and δ be as above. Then, we have $\delta_e \leq x_e(\mathcal{P}_r)$ for all edges e of G. Thus $z(\mathcal{Q}, \delta) \leq z(\mathcal{P}_r)$.*

We are now ready to complete the description and analysis of our approximation algorithm for csVPN. The input to the algorithm is a csVPN instance (G, b, c, f). In the proof below, we use OPT(.) to denote the cost of an optimal solution to the corresponding csVPN or ndCR instance.

THEOREM 3.4. *Algorithm* 1 *is a ρ-approximation algorithm for* csVPN.

Proof. From Corollary 2.6, we know that there exists a tree T such that $z(\mathcal{P}^T, \gamma^T) = \mathrm{OPT}(G, b, c, f)$. By Lemma 3.2, $\min_{r \in V(T)} z(\mathcal{P}_r^T) \leq z(\mathcal{P}^T, \gamma^T)$. Since $\mathcal{P}_r^T$ is a solution to the ndCR instance (G, r, b^r, c, f), it follows that $\min_{r \in V(T)} z(\mathcal{P}_r^T) \geq \min_{r \in V} \mathrm{OPT}(G, r, b^r, c, f)$. Let $\tilde{r} \in V$ be such that $\min_{r \in V} \mathrm{OPT}(G, r, b^r, c, f) =$

ALGORITHM 1. APPROXIMATION ALGORITHM FOR csVPN

(1) For each $r \in V$, compute a ρ-approximate solution $\mathcal{P}_r$ to the ndCR instance (G, r, b^r, c, f).
(2) Let r^* be such that $z(\mathcal{P}_{r^*}) = \min_{r \in V} z(\mathcal{P}_r)$.
(3) From $\mathcal{P}_{r^*}$, build a solution $(\mathcal{Q}, \delta)$ to the csVPN instance (G, b, c, f) as in Lemma 3.3.
(4) Output $(\mathcal{Q}, \delta)$.

$\mathrm{OPT}(G, \tilde{r}, b^{\tilde{r}}, c, f)$. By choice of r^*, $z(\mathcal{P}_{r^*}) \leq z(\mathcal{P}_{\tilde{r}}) \leq \rho\, \mathrm{OPT}(G, \tilde{r}, b^{\tilde{r}}, c, f)$. From Lemma 3.3, $z(\mathcal{Q}, \delta) \leq z(\mathcal{P}_{r^*})$. Putting everything together, we conclude $z(\mathcal{Q}, \delta) \leq \rho\, \mathrm{OPT}(G, b, c, f)$, as desired. ∎

By Theorem 3.1, there exists a ρ-approximation algorithm for csVPN with $\rho = 24.92$.

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 the same factor on the corresponding instances. In particular, if f is defined as $f(x) := \min\{\mu x, M\}$, for two positive numbers μ, M, then the ndCR instance constructed in Algorithm 1 from a csVPN instance is, except for decomposing into paths, just an instance of the so-called SSRB problem [14, 1]. 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 μ 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 et al. [1].

4. A remark on general concave functions. It is known (see, e.g., [13]) that the tree routing property is satisfied by every CR instance such that g is nondecreasing, and it follows from our results that this also holds when g is axis symmetric. A natural question arises: is the tree routing 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. Consider an instance (G, r, b, c, g) of the CR 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 $e = \{0, 1\}$, and $c_e := 0$ 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-axis-symmetric, 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 ϵ and M, respectively, small and big enough, every tree solution costs more.

Acknowledgment. We thank the three anonymous referees for providing remarks that guided us when revising this manuscript.

Note added in preparation. Following the results in this manuscript, an alternative proof of the fact that the tree routing property holds for csVPN has been given [5]. This proof, however, does not show that it also holds for sCR.

REFERENCES

[1] F. Eisenbrand, F. Grandoni, T. Rothvoss, and G. Schäfer, *Approximating connected facility location problems via random facility sampling and core detouring*, in Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2008), ACM, New York, 2008, pp. 1174–1183.

[2] T. Erlebach and M. Rüegg, *Optimal bandwidth reservation in hose-model VPNs with multi-path routing*, in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), IEEE, New York, 2004, pp. 2275–2282.

[3] J.A. Fingerhut, S. Suri, and J.S. Turner, *Designing least-cost nonblocking broadband networks*, J. Algorithms, 24 (1997), pp. 287–309.

[4] N. Goyal, N. Olver, and B. Shepherd, *The VPN conjecture is true*, in Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC 2008), ACM, New York, 2008, pp. 443–450.

[5] N. Goyal, N. Olver, and B. Shepherd, *private communication*, 2009.

[6] F. Grandoni and G. Italiano, *Improved approximation for single-sink buy-at-bulk*, in Proceedings of the 17th International Symposium on Algorithms and Computation (ISAAC 2006), Lecture Notes in Comput. Sci. 4288, Springer, Berlin, 2006, pp. 111–120.

[7] F. Grandoni, V. Kaibel, G. Oriolo, and M. Skutella, *A short proof of the VPN tree routing conjecture on ring networks*, Oper. Res. Lett., 36 (2008), pp. 361–365.

[8] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, *Provisioning a virtual private network: A network design problem for multicommodity flow*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001), ACM, New York, 2001, pp. 389–398.

[9] A. Gupta and A. Kumar, *private communication*, 2002.

[10] G.M. Guisewite and P.M. Pardalos, *Minimum concave-cost network flow problems: Applications, complexity, and algorithms*, Computational methods in global optimization, Ann. Oper. Res., 25 (1990), pp. 75–99.

[11] C.A.J. Hurkens, J.C.M. Keijsper, and L. Stougie, *Virtual private network design: A proof of the tree routing conjecture on ring networks*, SIAM J. Discrete Math., 21 (2007), pp. 482–503.

[12] G. Italiano, S. Leonardi, and G. Oriolo, *Design of trees in the hose model: The balanced case*, Oper. Res. Lett., 34 (2006), pp. 601–606.

[13] D.R. Karger and M. Minkoff, *Building Steiner trees with incomplete global knowledge*, in Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS 2000), IEEE Computer Society Press, Los Alamitos, CA, 2000, pp. 613–623.

[14] A. Gupta, A. Kumar, M. Pàl, and T. Roughgarden, *Approximation via cost sharing: Simpler and better approximation algorithms for network design*, J. ACM, 54 (2007), article 11.