Reserving Resilient Capacity in a Network

G. Brightwell
Centre for Discrete and Applicable Mathematics
London School of Economics
London, U.K.

G. Oriolo
Dept. of Computer Science, Systems and Production
University of "Tor Vergata"
Rome, Italy

F.B. Shepherd
Bell Laboratories
Lucent Technologies
Murray Hill, NJ, USA

DEDICATED TO THE MEMORY OF EWART LOWE

Abstract

We examine various problems concerning the reservation of capacity in a given network, where each arc has a per-unit cost, so as to be "resilient" against one or more arc failures. For a given pair (s, t) of nodes and demand T, we require that, on the failure of any k arcs of the network, there is sufficient reserved capacity in the remainder of the network to support an (s, t) flow of value T. This problem can be solved in polynomial time for any fixed k, but we show that it is NP-hard if we are required to reserve an integer capacity on each arc.

We concentrate on the case where the reservation has to consist of a collection of arc-disjoint paths: here we give a very simple algorithm to find a minimum cost fractional solution, based on finding successive shortest paths in the network. Unlike traditional network flow problems, the integral version is NP-hard: we do however give a polynomial time $\frac{15}{14}$-approximation algorithm in the case $k = 1$, and show that this bound is best possible unless P = NP.

Keywords: *Network flows, resilience, capacity reservation.*

A longer version of this paper appears as [11].

1 Introduction

A commonly encountered network design problem is that of reserving capacities in a network so as to support a given set of pairwise traffic demands. Algorithms for this *network capacity allocation problem* have been developed by a number of groups, see for example [6, 8, 9, 19, 21, 22, 23]. One significant drawback to this 'vanilla' capacity reservation model is that it does not account for the failure of certain network elements. For instance, if we simply reserve capacity for a commodity along a single path, we make ourselves totally vulnerable to the failure of any arc (or node) along this path. In many practical settings, this is not acceptable, and we thus wish to reserve our capacities so as to be resilient to certain failure states in the network.

Several groups have recently addressed this issue of "resilience" or "survivability" in network design problems, e.g., [2, 3, 5, 10, 14, 15, 20, 24, 25, 26, 27]. Their solution techniques are based primarily on polyhedral or branch and cut methods and hence produce exact optimal solutions if they terminate, and usually give some guarantee of optimality even before terminating. Such techniques are not always the right selection in a given scenario. On one hand, the need for exact solutions must be balanced with the degree to which the input costs and data are known or certain. These methods also do not exhibit polynomially bounded running time, and hence performance may not scale well as network sizes grow. This may prove to be an even larger issue for resilient capacity reservation which appears initially to be a much less tractable problem computationally (c.f. [10])). In addition, many applications require solutions in time scales which force the adoption of heuristics with fast run-time properties. Many existing network planning tools for the (vanilla) capacity allocation problem resort to some form of repeated single-source-destination heuristics. That is, for each demand pair (s_i, t_i) a shortest path is found and then as much flow as possible is pushed on this path. The process is then repeated until all demands are met. In its most general form, the cost of an edge is updated as a function of its remaining capacity. This approach is fast and allows for trivial implementation in software. In its simpler forms, however, one easily concocts examples where it produces solutions arbitrarily far from the optimum. Of course, there are many examples where the exact methods do not even find feasible solutions in a comparable running time. Moreover, they can require substantial mathematical sophistication on the part of its implementors.

Another situation where the single source-destination pair model applies is in an on-line setting. Here, the demands are being given sequentially as they arise in the network. This area is becoming increasingly important as network management becomes a concern of network operators. This is heightened by the changing nature of demands from customers. In particular, bursty or short-term data transfers are becoming increasingly common. As a result, larger amounts of point to point bandwidth are being traded on shorter time frames.

The present paper is dedicated to adapting the shortest path (or mincost flow) model to a *minimum cost resilient capacity reservation* model. We are restricting ourselves to the study of resilient capacity allocation for traffic generated by a single source-destination pair of nodes; we show that even this case presents some surprising difficulties. Overviews of previous computational and theoretical work on related survivability and augmentation problems can be found in [18] and [17].

We adopt the viewpoint that the network, with specified nodes s and t, is given to us, along with a per-unit cost c_a for each arc a, and that we are free to reserve, once and for all, as much capacity as we like on whatever arcs we choose. Our objective is to find a "reservation vector" x minimizing the *total cost*, $\text{cost}(x) = \sum_a c_a x_a$, subject to supporting a given target amount T of traffic from s to t, even if any one, or more generally any k, of the arcs in the network fails.

This rough description of the problem admits many different versions, depending on the type of network we are dealing with, the way we are required to recover from arc failures, and especially on any structure imposed on the vector of reserved capacities itself. In this paper, we consider two types of constraints on the capacity reservation vector.

1. Integrality: We may be forced to reserve capacity in discrete amounts, so that our reservation vector must be integral.

2. Structural: Specifically, it may be required that our reservation vector be formed by selecting a collection of disjoint (s,t)-paths (i.e., directed paths from s to t in the network), and assigning a capacity to each path – we call such a reservation a *diverse-path reservation.*

Diverse-path solutions have several features which are attractive to network planners. For a start, a diverse-path routing may be "hardwired" at the terminating nodes, thus decreasing routing complexity. If traffic flow control is centralized, then this allows load balancing of traffic over the collection of diverse paths. Even if this is not the case, as in a noncooperative network, the restoration phase is much simpler since an arc failure may be treated as a path failure, and traffic routed along the path may be shifted to the remaining non-failed paths. One final advantage is that they are conceptually simple to visualize and work with in any operational setting.

2 Summary of results

In this paper, we consider various versions of the resilience problem; we summarize these and our results in this section.

Throughout, we suppose (sometimes implicitly) that we are given a directed graph (network) $D = (V, A)$ with *node set* V and *arc set* A. We always assume that D comes with two nodes permanently fixed as the *source* s and the *destination* t. We also suppose that we are given a rational number T (usually an integer) representing the required traffic flow from s to t through the network D in the case of failure, and an integer k representing the maximum number of arcs that may fail. Finally, we are also given a vector (c_a) of non-negative rational (again, usually integer) costs on the arcs a of D. We are seeking a *reservation vector*, which is a non-negative vector (x_a) on the arcs a, representing the amount of capacity reserved on the arcs of D.

A reservation vector $x = (x_a)$ is (T,k)*-resilient* if, for each set K of at most k arcs in A, the reserved capacities on the arcs in $A - K$ are sufficient to admit an (s,t) flow of value T. The *cost* of a reservation vector x is $\mathrm{cost}(x) = \sum_{a \in A} c_a x_a$. Our aim in all versions considered is to find a minimum cost (T, k)-resilient reservation vector x in D.

The problem stated above is the GENERAL RESILIENCE problem. We do not normally treat k as part of the input, so an instance of this problem, for resilience against k failures, consists of a network D, with specified source and destination, a demand T, and a cost vector c on the arcs of D. If the reservation vector x is required to be an integer, then the demand T and all the costs must also be integers, and we have the INTEGER RESILIENCE problem.

As explained in Section 3, for each fixed k, GENERAL RESILIENCE can be solved in polynomial time using linear programming or the ellipsoid algorithm. However, we show in Section 7 that INTEGER RESILIENCE is strongly NP-hard even for $k = 1$, although we do give a simple $(k + 1)$-approximation algorithm.

For most of the paper, we concentrate on the case where the reservation vector is also required to be a *diverse paths reservation*, i.e., it is derived from a set $P_1, \ldots, P_m$ of arc-disjoint (s, t)-paths, with capacity x_j reserved on each arc of path P_j $(j = 1, \ldots, m)$.

In Section 4, we consider the case where the (s, t)-paths $P_1, \ldots, P_m$ to be used are prespecified. In this case, we may as well consider each path as a single arc from s to t, whose cost is the sum of the individual arc-costs, and we have the k-FAILURE ALLOCATION problem, an instance of which consists simply of a demand T and a sequence of costs $c_1, \ldots, c_m$, where we assume that $c_1 \leq \cdots \leq c_m$. The problem is to find a non-negative real vector $x = (x_1, \ldots, x_m)$ to minimize $\mathrm{cost}(x) = \sum c_j x_j$ subject to the (T, k)-resilience constraint that the sum of any $m - k$ of the x_j is at least T. If we additionally impose the constraint that the x_j be integers, then we have the INTEGER k-FAILURE ALLOCATION problem.

We give an extremely simple algorithm for solving both k-FAILURE ALLOCATION and its integer counterpart. For the case $k = 1$, a polynomial time algorithm (for the integer version) based on convex optimization is already explained in the work of Bartholdi et al [7] which addresses certain integer programming problems arising from matrices with the circular ones properties in each row. Our results provide additional information on the structure of the optimal fractional and integral solutions which is needed later in the paper when we study the INTEGER DIVERSE-PATH RESILIENCE problem without the paths being prefixed. Some of these structural properties were independently observed by Bienstock and Muratore [10] who gave a complete linear description for an associated polyhedron.

In Sections 5 and 6, we turn to the case when our diverse-path reservation x may use any set of diverse (s, t)-paths. The problems we consider are the DIVERSE-PATH RESILIENCE problem and INTEGER DIVERSE-PATH RESILIENCE problem, which are exactly the same as GENERAL RESILIENCE and INTEGER RESILIENCE except that the reservation vector x is required to be a diverse-path reservation.

Using the information from Section 4 about the nature of any optimal diverse-path reservation, we give a simple combinatorial algorithm, based on finding successive shortest paths in the network, to solve DIVERSE-PATH RESILIENCE.

However, INTEGER DIVERSE-PATH RESILIENCE turns out to be NP-hard, even for $k = 1$. Here instead we give a polynomial time $\frac{15}{14}$-approximate algorithm in the case $k = 1$, and show this bound is the best possible (if P $\neq$ NP). Similar results hold if k takes other values, or is unrestricted.

We conclude the paper with a discussion of a possible application to the case of more than one source-destination pair, and a few remarks about other types of resilience problems.

3 Polyhedral Formulation

Given a directed graph $D = (V, A)$, and any $S \subseteq V$, let $\delta^+(S)$ denote the set of arcs with tail in S and head in $V - S$, and set $\delta^-(S) = \delta^+(V - S)$. We call $S \subseteq V$ an *(s,t)-set* if $s \in S$ and $t \in V - S$.

Let $\mathbb{Q}_+$ denote the set of non-negative rational numbers, so that $\mathbb{Q}_+^A$ is the set of all assignments of non-negative rationals to each member of the arc-set A, which we frequently view as a vector. For any vector $x \in \mathbb{Q}_+^A$ and $A' \subseteq A$, we denote by $x(A')$ the sum $\sum_{a \in A'} x_a$.

The problem of finding a minimum cost (T,k)-resilient reservation vector can be expressed as an optimization problem over a certain polyhedron, which we now describe.

For any rational T, (s,t)-set S, and set $K \subseteq \delta^+(S)$ of at most k arcs, the *partial T-cut constraint* associated with the pair (S, K) is the constraint $x(\delta^+(S) - K) \geq T$. The *resilience polyhedron* is defined by the system of all partial cut constraints.

$$\mathcal{R}(T,k,D) = \left\{ \begin{array}{lll} x \in \mathbb{Q}_+^A & : & x(\delta^+(S) - K) \geq T \quad \text{for each } (s,t)\text{-set } S \\ & & \qquad\qquad\qquad\quad \text{and } K \subseteq \delta^+(S) \text{ with } |K| \leq k \end{array} \right\}. \tag{1}$$

Note that $\mathcal{R}(T,k,D)$ is empty if there is an (s,t)-set S with $\delta^+(S)$ of size at most k, and otherwise $\mathcal{R}(T,k,D)$ is full-dimensional. It is straightforward to verify that $\mathcal{R}(T,k,D)$ consists exactly of the (T,k)-resilient vectors, and so the General Resilience problem – finding a minimum cost (T,k)-resilient reservation – is that of minimizing the linear function $\text{cost}(x) = \sum_{a \in A} c_a x_a$ over $\mathcal{R}(T,k,D)$.

A consequence of this formulation is that, for each fixed k, there is a polynomial time algorithm to solve General Resilience. Indeed, it is easily seen that the separation problem for $\mathcal{R}(T,k,D)$ amounts to solving at most $|A|^k$ maximum flow problems. Moreover the problem can be rephrased as that of finding a single edge capacity vector x, together with an (s,t) flow vector y^K of value T for each failing set K of size k (i.e., with $y^K_a = 0$ for $a \in K$), subject to the constraint $y^K_a \leq x_a$ for each arc a and each failing set K. This formulation constitutes a linear program with a polynomially bounded number of variables and constraints. This is not, however offered as a practical approach, even for $k = 1$, and the remainder of the paper addresses the task of finding more direct combinatorial algorithms.

4 Reservations on a Fixed Set of Paths

A version of the material in this section, written for a non-technical audience, appears as [13]. A more thorough handling of the polyhedron considered implicitly herein (including a complete linear description of the integer hull) has been given by Bienstock and Muratore [10]. The case $k = 1$ can also be regarded as a special case of a problem treated by Bartholdi, Orlin and Ratliff [7]; our methods give a somewhat simpler solution in this special case.

We start by considering k-FAILURE ALLOCATION and its integer version. Recall that these problems can be formulated as follows.

k-FAILURE ALLOCATION
Given a demand T, and a sequence of non-negative costs $c_1 \leq c_2 \leq \ldots \leq c_m$, find a non-negative real vector $x = (x_1, x_2, \ldots, x_m)$ minimizing $\text{cost}(x) = \sum_{i=1}^{m} c_i x_i$, subject to the constraint that $\sum_{i \notin K} x_i \geq T$ for every set K of size k.

INTEGER k-FAILURE ALLOCATION
As above, with x required to be integer.

We start with a result that is fundamental in much of the rest of the paper. For $k < j \leq m$, let $z^{j,k}$ be the reservation vector defined by

$$z_i^{j,k} = \begin{cases} T/(j-k) & i \leq j \\ 0 & i > j. \end{cases}$$

Theorem 4.1 *An optimal solution to k-*FAILURE ALLOCATION *is obtained at one of the solutions $z^{j,k}$.*

Proof. Because of the symmetry of the situation and the ordering of the costs c_i, it is clear that there is an optimal solution such that $x_1 \geq x_2 \geq \ldots \geq x_m$. Thus we lose nothing by including these inequalities as constraints. Once we do this, we see that, if the constraint $x_{k+1} + x_{k+2} + \ldots + x_m \geq T$ is satisfied, all the other resilience constraints given by the removal of k of the paths are automatically satisfied. Thus we may reformulate the problem as follows.

Given a demand T, and a sequence of costs $c_1 \leq c_2 \leq \ldots \leq c_m$, find non-negative real numbers $x_1, \ldots, x_m$ to minimize $\sum_i c_i x_i$, subject to the constraints $x_1 \geq x_2 \geq \ldots \geq x_m \geq 0$ and $x_{k+1} + \ldots + x_m \geq T$.

We note for future reference that the same reformulation goes through if the x_i are all constrained to be integers.

Consider a basic optimal solution for the resulting linear program which necessarily satisfies m linearly independent inequalities with equality. If there are j non-zero variables at the optimum, then the only possibility is that all of the inequalities $x_1 \geq x_2, \ldots, x_{j-1} \geq x_j$, $x_{j+1} \geq \ldots \geq x_m \geq 0$ and $x_{k+1} + \ldots + x_m \geq T$ are satisfied with equality, i.e., that $x_1 = x_2 = \ldots = x_j = T/(j-k)$ and $x_{j+1} = \ldots = x_m = 0$. This is just the solution $z^{j,k}$ and the result follows. □

Solving k-FAILURE ALLOCATION thus amounts to choosing amongst the solutions $z^{j,k}$. In fact the structure of the problem allows a particularly simple procedure for doing this. Note that $\frac{1}{T}\text{cost}(z^{j,k})$ is equal to $A_{j,k} = (c_1 + \ldots + c_j)/(j-k)$, which is the average of $c_1 + c_2 + \ldots + c_{k+1}$,

$c_{k+2}, \ldots,$ and c_j. The $A_{j,k}$ are decreasing in j up to the minimum, which is attained for the last j where $c_j < A_{j-1,k}$, and increasing thereafter: this unimodality property will be a recurring theme. Thus we may terminate our such for the minimum value of $\mathrm{cost}(z^{j,k})$ if ever we find that $c_{j+1} \geq A_{j,k}$.

For INTEGER k-FAILURE ALLOCATION, we show that the following procedure suffices.

First find the optimal fractional solution $z^{j,k}$, i.e., the optimal solution of the corresponding instance of k-FAILURE ALLOCATION. Then (if $z^{j,k}$ is not already an integer vector) consider the two integer solutions "nearest" to $z^{j,k}$, as follows.

(a) Set r equal to either $\lfloor T/(j-k) \rfloor$ or $\lceil T/(j-k) \rceil$. (Here $\lceil a \rceil$ denotes the next integer above the real number a, and $\lfloor a \rfloor$ the next integer below.) Note that r may be zero if $T < j - k$.

(b) If r is one of the two chosen values and r is nonzero, we attempt to construct a solution x with all the non-zero x_i, except possibly one, equal to r. To do this, we set $\ell = \lceil T/r \rceil + 1$; if $\ell \leq m$, set $x_1 = x_2 = \ldots = x_{\ell-1} = r$, $x_\ell = T - (\ell - k - 1)r$, and $x_{\ell+1} = \ldots = x_m = 0$. (The choice of ℓ ensures that $x_\ell \leq x_1 = r$. If $r = \lfloor T/(j-k) \rfloor$, we could have $\ell > m$, but this is not possible with $r = \lceil T/(j-k) \rceil$.)

Note that this is a feasible solution, since removing any k of the x_i leaves capacity at least $(\ell - k - 1)r + x_\ell$, which is constructed to be at least T.

(c) We now have either one or two candidate integral solutions, corresponding to the two choices of r in (a). We denote by $z^{j,k,+}$ the solution with $r = \lfloor T/(j-k) \rfloor$ (if it is feasible), and by $z^{j,k,-}$ the solution with $r = \lceil T/(j-k) \rceil$ (which is always feasible)[1]. To finish, just calculate the costs of the two solutions, and choose the lower.

Theorem 4.2 *Suppose we have an instance of* k-FAILURE ALLOCATION *in which the optimal solution is* $z^{j,k}$. *Then the optimal solution* x *to the corresponding instance of* INTEGER k-FAILURE ALLOCATION *is either* $x = z^{j,k,+}$ *or* $x = z^{j,k,-}$.

Proof. We work with the reformulation of the problem as in the beginning of the proof of Theorem 4.1, which, as we noted, is also valid for the integer case. Suppose that x is an optimal integer solution.

Clearly we have $x_1 = x_2 = \ldots = x_{k+1}$ at the optimum. Now suppose that some x_j is non-zero, but that not all of $x_1, \ldots, x_{j-1}$ are equal. Then let i be the minimum index with $x_i < x_1$; note that $i \geq k+2$ by the previous observation. Also let x_ℓ be the last non-zero variable, so $k + 2 \leq i < j \leq \ell$. Increasing x_i by one and decreasing x_ℓ by one keeps the solution feasible, and $x_{k+1} + \ldots + x_m$ is unaltered. Also, this operation does not increase the cost.

We have thus shown that one may restrict attention to integral solutions where there is some j, with $k + 2 \leq j \leq m$, such that all of $x_1, \ldots, x_{j-1}$ are equal, and all of $x_{j+1}, \ldots, x_m$ are equal to 0. If the value of x_j is q, then the common value of the earlier x_i is $(T-q)/(j-k-1)$, which is an integer, at least q.

[1]The notation $z^{j,k,+}$, $z^{j,k,-}$ indicates moving from $z^{j,k}$ towards $z^{j+1,k}$ or $z^{j-1,k}$.

At this point, there is still potentially one solution for each integer value $x_1 \geq T/m$, namely to set $j = k + \lceil T/x_1 \rceil$, and $q = T - (j-k-1)x_1$; observe that this solution is equal to

$$\frac{q(j-k)}{T} z^{j,k} + \left(1 - \frac{q(j-k)}{T}\right) z^{j-1,k}.$$

Thus each of our candidate integral solutions is a convex combination of two consecutive $z^{i,k}$s. Let $\mathcal{A}$ be the set of all such convex combinations; we think of $\mathcal{A}$ as a "path" with vertices corresponding to $z^{k+1,k}, \ldots, z^{m,k}$; any vector on this path gives a feasible solution. Note that the first coordinate value x_1 decreases along $\mathcal{A}$, and the solution cost is unimodal along $\mathcal{A}$, since it is linear between the vertices. If the fractional optimum is attained at a vertex with x_1 equal to r, then the lowest cost integer solution on $\mathcal{A}$, and hence the overall integer optimum, is obtained by taking x_1 to be either $\lceil r \rceil$ or $\lfloor r \rfloor$. This amounts to taking either $z^{j,k,+}$ or $z^{j,k,-}$, as required. □

We close this section with a bound relating the costs of the optimal integral and fractional solutions to k-Failure Allocation.

Proposition 4.3 *For any $j > k \geq 1$, $\mathrm{cost}(z^{j,k,-})/\mathrm{cost}(z^{j,k}) < 1 + \frac{k(j-k)}{jT} < 1 + \frac{k}{T}$.*

Proof. Recall that, in the solution $z^{j,k}$, the j cheapest paths are chosen, each with capacity $T/(j-k)$. The "rounded" solution $z^{j,k,-}$ is obtained from this by taking the $\ell - 1$ cheapest paths with capacity $x = \lceil T/(j-k) \rceil$, and the next cheapest path with capacity $T - (\ell-k-1)x$, where $\ell = \lceil T/x \rceil + k \leq j$.

The first observation is that the average cost per unit of reservation in $z^{j,k,-}$ is no greater than that in $z^{j,k}$. Thus $\mathrm{cost}(z^{j,k,-})/\mathrm{cost}(z^{j,k})$ is at most the ratio of the total numbers of units of capacity reserved in the two allocations.

In $z^{j,k}$, a total of $jT/(j-k)$ units of capacity are allocated, while in $z^{j,k,-}$, the total is $(\ell-1)x + T - (\ell-k-1)x = T + kx$. Hence we have

$$\frac{\mathrm{cost}(z^{j,k,-})}{\mathrm{cost}(z^{j,k})} \leq \frac{T + kx}{jT/(j-k)} = 1 + \frac{k}{jT}\left((j-k)x - T\right).$$

Now $(j-k)x - T < j - k$, by definition of x, so we have the estimate as claimed. □

Corollary 4.4 *If O_I is the cost of the optimal solution to an instance of* Integer k-Failure Allocation *with target flow T, and O_F the cost of the optimal solution to the corresponding instance of* k-Failure Allocation, *then $O_I < (1 + k/T)O_F$.*

Consideration of the proof of Proposition 4.3 shows that the ratio $(1 + k/T)$ cannot in general be improved. Indeed, if our network consists of a large number M of paths of cost 1, then it is easy to see that $O_I = T + k$, whereas $O_F = MT/(M-k)$; as $M \to \infty$, $O_I/O_F \to (1 + k/T)$.

5 Diverse-Path Reservations without Specified Paths

We now turn to the Diverse-Path Resilience problem, when we are still required to find a resilient reservation consisting of a set of diverse paths in a given network D, but we are not restricted as to what paths we may use. Our aim here is to give a fast algorithm for Diverse-Path Resilience, whatever number k of failures is to be allowed for.

Theorem 4.1 implies that the optimal solution has as support the arcs of some $j > k$ diverse paths, with each arc in the support given capacity $T/(j-k)$. Of course, j can only take integer values up to the (s,t)-connectivity $\kappa = \kappa(D)$ of D. We may take advantage of this structure and apply the *successive shortest path method* – c.f. [1] – for minimum cost flow problems, thus only needing to solve $\kappa(D)$ shortest path problems.

For an arc $a = (u, v) \in A$, we let a^- denote an "artificial" arc (v, u), not present in D. In the course of the following algorithm, we construct a series of auxiliary digraphs D_j, each of which contains exactly one from each pair a, a^-. We assume that we are given a digraph D with $\kappa(D) > k$.

In the algorithm Paths below, we find a succession of arc-sets $\mathcal{P}_1, \mathcal{P}_2, \ldots$, where each $\mathcal{P}_j$ is the arc-set of a set of j diverse (s,t)-paths of minimum cost. Each $\mathcal{P}_{j+1}$ is derived from $\mathcal{P}_j$ by adding a cheapest path in the network D_j with costs c^j. Adding the arc a^- corresponds to removing the arc a. In line with our earlier notation, $z^{j,k}$ denotes the diverse-path reservation using the paths of $\mathcal{P}_j$. The algorithm terminates if $z^{j+1,k}$ is at least as expensive as $z^{j,k}$.

```
PATHS(D, k)
{
    j = 0; D_0 = D; c^0 = c; P_0 = ∅;
    While (D_j contains a directed (s,t)-path)
        Let Q_j be the arc-set of a minimum c^j-cost directed (s,t)-path in D_j
        Set P_{j+1} = (P_j − R) ∪ F
            where R = {a ∈ A : a^− ∈ Q_j}
            and F = A ∩ Q_j
        If j ≥ k, let z^{j+1,k} be the vector obtained by assigning
            T/(j + 1 − k) to each arc in P_{j+1}
        If j ≥ k and cost(z^{j+1,k}) ≥ cost(z^{j,k})
            then Output(z^{j,k}) and Quit
        D_{j+1}, c^{j+1} are the same as D_j, c^j except
            if a ∈ R
                remove a^−, and include a with cost c_a^{j+1} = c_a
            if a ∈ F
                remove a, and include a^− with cost c_{a^−}^{j+1} = −c_a
        Set j = j + 1;
    EndWhile
    Output(z^{j,k})
}
```

We also refer to the version of the algorithm which does not terminate early and thus generates a reservation vector $z^{j,k}$ for every $j = k+1, \ldots, \kappa(D)$.

Proposition 5.1 (c.f. [1]) *Let c be a nonnegative vector of arc costs in a network $D = (V, A)$. The algorithm* PATHS *finds a minimum cost (T, k)-resilient diverse-path reservation.*

To establish correctness, we need two facts. First, for each j, the collection $\mathcal{P}_j$ induces a minimum cost collection of j diverse (s,t)-paths; this follows from the correctness of the successive shortest path method. This implies that each solution $z^{j,k}$ is the minimum cost solution using j paths, and hence the minimum cost (T, k)-resilient vector is among these vectors $z^{j,k}$. Moreover, traditional flow theory implies that for each $j \geq k+1$, $z^{j,k}$ is a minimum cost flow of value $jT/(j-k)$ subject to the capacities $T/(j-k)$ on each arc. Second, as we now show, the sequence $\mathrm{cost}(z^{j,k})$ is unimodal for $j \geq k+1$, and so early termination is justified.

Proposition 5.2 *Let h, i and j be such that $k < h < i < j \leq \kappa(D)$. If $\mathrm{cost}(z^{h,k}) \leq \mathrm{cost}(z^{i,k})$ then $\mathrm{cost}(z^{i,k}) \leq \mathrm{cost}(z^{j,k})$.*

Proof. Suppose the contrary: there exists h, i, j, with $h < i < j$ such that $\mathrm{cost}(z^{h,k}) \leq \mathrm{cost}(z^{i,k})$ and $\mathrm{cost}(z^{i,k}) > \mathrm{cost}(z^{j,k})$. Let $M = T\frac{i}{i-k}$ and choose $\lambda \in (0,1)$ such that $\frac{i}{i-k} = \lambda\frac{h}{h-k} + (1-\lambda)\frac{j}{j-k}$. Then $z' = \lambda z^{h,k} + (1-\lambda)z^{j,k}$ is a flow of value $iT/(i-1)$ and does not exceed $T/(i-1)$ on any arc. Thus by the remarks preceding the proposition, $\mathrm{cost}(z') \geq \mathrm{cost}(z^{i,k})$. But of course $\mathrm{cost}(z') = \lambda\mathrm{cost}(z^{h,k}) + (1-\lambda)\mathrm{cost}(z^{j,k}) < \mathrm{cost}(z^{i,k})$, a contradiction. □

6 Integer Diverse-Path Reservations

We now turn to INTEGER DIVERSE-PATH RESILIENCE, where we are required to find a minimum-cost diverse-path reservation taking integer values. We assume throughout this section that the demand T is also an integer.

Again the results of Section 4 give us information about the structure of an optimal solution: Theorem 4.2 shows that the support of an optimal solution consists of a collection of diverse (s,t)-paths $P_1, P_2, \ldots, P_j$ where the arcs of the first $j-1$ paths will reserve a common amount, r, of capacity, and the last path's arcs will reserve capacity $T - (j-1-k)r \leq r$.[2] We now show that the subproblem with $k = 1$ and $T = 3$, denoted by 3-IDP, is NP-hard. Let 2DIV-PATHS denote the problem of determining whether a given digraph D, with four distinct nodes s_1, t_1, s_2, t_2, contains a pair of arc-disjoint paths P_1, P_2, where P_i joins s_i and t_i $(i = 1, 2)$. Fortune, Hopcroft and Wyllie [16] show that this problem is NP-complete.

Theorem 6.1 *The problem* 3-IDP *is NP-hard. Furthermore, unless $P = NP$, there is no polynomial time $(1+\varepsilon)$-approximation algorithm with $\varepsilon < \frac{1}{14}$.*

[2] In essence, we thus need to solve an integer 2-multicommodity flow problem where both commodities have the same source and destination.

Proof. Suppose that we are given an instance of 2DIV-PATHS as above. Construct a digraph obtained from D by adding new nodes s, t as well as the arcs (s, t), (s, s_1), (s, s_2), (t_1, t) and (t_2, t) with costs 3, 1, 2, 1 and 2 respectively. All remaining arcs have cost zero. This is our instance of 3-IDP. From Theorem 4.2, we deduce that an optimal 3-resilient reservation on diverse paths will either have support on (i) 2 diverse paths, in which case capacity 3 is reserved on each of the arcs of these paths, or (ii) 3 diverse paths in which case two of the paths will have reserved capacity 2 and the third capacity 1.

Note that the cheapest collection of 2 diverse paths has cost 5 and hence any solution of the form (i) will have cost at least 15. Next note that if there exists a positive solution to the instance of 2DIV-PATHS, with P_i a path between s_i, t_i $(i = 1, 2)$, then by assigning capacity 2 to the arc (s, t) and the arcs of P_1, and capacity 1 to the arcs of P_2 we obtain a solution to 3-IDP of cost 14. Conversely, if the instance of 2DIV-PATHS has no solution, then any "3-path" solution to 3-IDP will use only paths of cost 3, from which we deduce that the reservation will cost at least 15. Thus the optimal solution to the instance of 3-IDP is 14 if and only if the instance of 2DIV-PATHS has a positive solution, and is otherwise at least 15. The result follows. □

We continue to concentrate on the case $k = 1$, but allow T to take arbitrary integer values. We have already seen in Proposition 4.3 that applying the rounding procedure to an optimal fractional solution yields a $(1+\frac{1}{T})$-approximation to the optimal fractional solution, and hence to the optimal integral solution. It is clear that the optimal fractional solution is integral in the case $T = 2$, so in fact we have a polynomial time $\frac{4}{3}$-approximation algorithm for arbitrary T: we now improve this to a $\frac{15}{14}$-approximation algorithm which, in view of Theorem 6.1, is best-possible. Note that, by Proposition 4.3, we may assume that $T \leq 13$.

Consider the polynomial time algorithm $\mathcal{A}$ (based on PATHS) that finds, for each value of j, the fractional solution $z^{j,1}$ based on *some* cheapest set of j diverse paths, and the two "rounded" integer solutions $z^{j,1,-}$ and $z^{j,1,+}$, and chooses the best among all of the integer solutions. The algorithm $\mathcal{A}$ can fail to find the optimal integer solution because it may use a minimum cost set of ℓ paths in which the costs are distributed "more evenly" between the paths (in particular, the most expensive path is cheaper) than in some other (not necessarily even minimum cost) set of ℓ paths. All we know is that an optimal solution for an instance of INTEGER DIVERSE-PATH RESILIENCE has the *same form* as either $z^{\ell,1,-}$ or $z^{\ell,1,+}$ for some ℓ, since it arises from a similar rounding process applied to *some* collection of diverse paths.

Let O_F be the cost of a fractional optimum solution, O_I the cost of an integer optimum solution, and O_A the best solution among those considered by the algorithm, i.e., the value returned by $\mathcal{A}$. Clearly we have $O_F \leq O_I \leq O_A$.

Theorem 6.2 *The algorithm $\mathcal{A}$ is a $\frac{15}{14}$-approximate algorithm for* INTEGER DIVERSE-PATH RESILIENCE *with $k = 1$, that is, for each instance: $O_A \leq \frac{15}{14} O_I$.*

We note that the quality of approximation by the algorithm depends greatly on the input T. If we view $\mathcal{A}$ as an infinite collection of algorithms $\{\mathcal{A}_T\}_{T=1}^{\infty}$, each restricted to instances with a fixed value of the demand T, then many of these – in particular, those with large values of T – are $(1+\epsilon)$-approximate algorithms with $\epsilon < \frac{1}{14}$. Indeed, Proposition 4.3 tells us that, for each T, $O_A < (1+1/T)O_F \le (1+1/T)O_I$.

Furthermore, we note in the course of the proof that, for $T = 1, 2, 4, 6$ and 12, $\mathcal{A}$ solves the problem exactly.

Proof. Take any instance of the problem, and let z^* be an optimum solution, say using paths $P_1, \ldots, P_\ell, P_{\ell+1}$. We certainly know that there is no better solution using *these* paths, so by Theorem 4.2 we know that $P_1, \ldots, P_\ell$ all have the same reserved capacity r under z^*, and path $P_{\ell+1}$, which has the greatest cost among these paths, has reserved capacity $y \le r$. Furthermore, r is either $\lfloor T/(\ell-1) \rfloor$ or $\lceil T/\ell \rceil$, with $T = (\ell-1)r + y$.

Clearly we can assume that $O_I \neq O_A$. In particular, this means that $0 < y < r$, otherwise one of $z^{\ell+1,1}$ or $z^{\ell,1}$ would be an integral solution found by $\mathcal{A}$ whose cost was at most that of z^*. We may thus assume that $\ell \ge 2$ and that $2 \le r \le (T+1)/2$.

The values of T and r determine y and ℓ. Also, as noted above, Proposition 4.3 implies the result for $T \ge 14$, so we may assume that $T \le 13$ and that $r \le \lceil T/2 \rceil \le 7$. There are thus only a finite number of possible forms of z^* (in fact, just 23 pairs (T, r) satisfy all the restrictions mentioned so far), and we rule all of these out using the same basic method. At this point, let us observe that there are no cases with $T = 1, 2, 4, 6$ or 12; for these values of T, any r not dividing T exactly is not of the form $\lfloor T/(\ell-1) \rfloor$ or $\lceil T/\ell \rceil$ for any integer ℓ. In particular, we may assume that we have $T \ge 3$.

We require a lower bound on $\text{cost}(z^*) = O_I$. Notice that z^* can be written as y times the characteristic vector of some set of $\ell+1$ diverse paths, plus $(r-y)$ times the characteristic vector of some set of ℓ diverse paths. Let C_i be the cost of reserving one unit of capacity on the arcs in a cheapest collection of i diverse paths. So we have $O_I = \text{cost}(z^*) \ge yC_{\ell+1} + (r-y)C_\ell$.

Our algorithm $\mathcal{A}$ considers some integer solution $z^\dagger$ of the same form as z^* (i.e., the same values of T, r, y, ℓ), using some set of $\ell+1$ paths of cost $C_{\ell+1}$. The cost of $z^\dagger$ is at most what it would be if all the $\ell+1$ paths had the same cost $(\ell r + y)C_{\ell+1}/(\ell+1)$. So O_A is at most this quantity, i.e.,

$$C_{\ell+1} \ge \frac{\ell+1}{\ell r + y} O_A = \frac{\ell+1}{T+r} O_A.$$

We aim for a similar bound on C_ℓ, and to get this we need to look at a solution produced by $\mathcal{A}$ on at most ℓ paths. Accordingly, let $r' = \lceil T/(\ell-1) \rceil$; there is some integer solution with reserved capacity r' on the first $m < \ell$ paths from C_ℓ, and $v \le r'$ on one further path, with total of reserved capacities on all the paths equal to $T + r'$. Our algorithm will have looked at an integer solution with a cost at least as low as some solution of this form, and the average cost of a path in any solution of this form is at most C_ℓ/ℓ, as in the proof of Proposition 4.3.

So we have $O_A \le (T+r')C_\ell/\ell$, i.e., $C_\ell \ge \frac{\ell}{T+r'} O_A$. We conclude that

$$O_I \ge \left(\frac{y(\ell+1)}{T+r} + \frac{(r-y)\ell}{T+r'} \right) O_A.$$

After a little manipulation, this becomes

$$\frac{O_I}{O_A} \geq 1 - \frac{(r'-r)(r-y)\ell}{(T+r)(T+r')} \equiv 1 - G.$$

We could now run through all the 23 cases separately and show that $G \leq 1/15$ in each case, but we can save a little effort.

First, we consider all cases with $\ell = 2$. In this case, we have $r' = T$, and $r = \lceil T/2 \rceil$, which implies that $r' = T = 2y+1$ and $r = y+1$. Now $G = 2y/(3y+2)(4y+2) \leq 1/15$ for $y \geq 1$.

Next, if $r = 2$, we have $y = 1$, $T = 2\ell - 1$ and $r' = 3$. This gives $G = \ell/(2\ell+1)(2\ell+2) \leq 1/15$ for $\ell \geq 2$. Assume from now on that $\ell, r \geq 3$. This implies that $T \geq 7$ and that $r < T/2$.

If $r' = r+1$ then, since also $(r-y)\ell < r\ell < T + r$, we have $G < 1/(T+r+1)$, and we are done if $T + r \geq 14$. So we may assume $T \leq 10$, whence, since $\ell \geq 3$, we have $r \leq 4$. On the other hand, if $r' = \lceil T/(\ell-1) \rceil \geq r+2 \geq (T+1)/\ell + 2$, then we have $T \geq \ell^2 + \ell - 1$; thus the only two cases with $T \leq 13$ are $\ell = 3$ and $T = 11, 13$.

From here on in, there seems to be no great saving on dealing with all the cases individually. Here are all the cases not so far ruled out.

T	r	ℓ	y	r'	G
7	3	3	1	4	6/110
8	3	3	2	4	3/132
9	4	3	1	5	9/182
10	3	4	1	4	8/182
10	4	3	2	5	6/210
11	4	3	3	6	6/255
13	5	3	3	7	12/360

All the values for G above are less than 1/15, so the theorem is proved. □

It is clear that this technique can be used to prove similar results for other values of k. If k is allowed to take any value, we have the following result.

Theorem 6.3 *There is a polynomial time $\frac{6}{5}$-approximation algorithm for* Integer Diverse-Path Resilience*, with k as part of the input. Furthermore, there is no polynomial time $(1+\epsilon)$-approximation algorithm for the problem with $\epsilon < \frac{1}{5}$ unless P = NP.*

Proof. (Sketch) Our algorithm is the obvious adaptation of algorithm $\mathcal{A}$: find optimal fractional solutions $z^{j,k}$ for each possible j, round these solutions, and choose the best. We define O_F, O_I and O_A as before.

An argument exactly as in Theorem 6.2 shows that, if O_I and O_A are not equal, then

$$\frac{O_I}{O_A} \geq 1 - \frac{(r'-r)(r-y)\ell k}{(T+rk)(T+r'k)} \equiv 1 - G,$$

where $r' = \lceil T/(\ell - k) \rceil$, r is equal to either $\lfloor T/(\ell - k) \rfloor$ or $\lceil T/(\ell - k + 1) \rceil$, and $T + rk = \ell r + y$, with $0 < y < r$. So we need to show that the quantity G is at most $1/6$ in all cases.

Note that $T + rk > \ell r$ and $T + r'k > r'k$, so we have

$$G \leq \frac{(r' - r)(r - y)}{rr'}.$$

For both possible values of r, one can show that the integers r', r and $(r' - r)(r - y)$ satisfy $r' > r > (r' - r)(r - y)$, so G is at most $n/(n + 1)(n + 2)$ for some non-negative integer n, and this quantity is always at most $1/6$, as required.

To see that this approximation ratio is best possible, take an instance I of 2DIV-PATHS, and a large value of k; set $T = 3$, and construct an instance of INTEGER DIVERSE-PATH RESILIENCE as follows. There are nodes a_i, b_i, c_i, d_i, for $i = 1, \ldots, k + 1$, as well as s and t. There are arcs of cost 1 from s to each a_i, from c_i to b_{i+1} ($i = 1, \ldots, k$), and from each d_i to t. Also there are arcs of cost 2 from s to b_1 and from c_{k+1} to t. We also take a copy of our instance I of 2DIV-PATHS for each i, with initial nodes a_i and b_i and corresponding terminal nodes d_i and c_i: all arcs involved cost 0. If there are arc-disjoint paths linking each a_i and d_i, and each b_i and c_i, then we can use these to make $k + 1$ diverse paths of cost 2, and another path (through all the b_i and c_i), of cost $k + 4$. Reserving 2 units on the first $k + 1$ paths and 1 unit on the last gives a $(3, k)$-resilient reservation of cost $5k + 8$. If there are no such linking paths, then there is no reservation costing less than $6k + 6$ (we can either reserve capacity 3 on each of the $k + 1$ diverse paths of cost 2, or use the only set of $k + 2$ diverse paths, in which each path has cost 3). □

7 Integer Resilience

We next show that the problem of finding a minimum cost integral (T, k)-resilient vector is NP-hard, even in the single-failure case $k = 1$. We couch the problem as a decision problem.

INTEGER RESILIENCE

Instance: a digraph D, with integer costs c_{ij} on the arcs, with a single source s and destination t, a demand T (integer), and a target cost C (integer).

Question: is there an integer reservation vector x on the arcs of D such that $c \cdot x \leq C$, and such that x is $(T, 1)$-resilient?

Theorem 7.1 INTEGER RESILIENCE *is strongly NP-complete.*

Proof. (We omit some of the details: a full proof can be found in [11].)

Certainly the problem is in NP, since checking $(T, 1)$-resilience simply involves finding flows of value T in the networks obtained by removing individual edges.

To prove the problem is NP-complete, we give a reduction from 3D-MATCHING. Recall that an instance of 3D-MATCHING consists of three sets A, B, C of size n, and a collection $\mathcal{T}$ of m "triangles" each containing exactly one element from each of A, B and C; the question is whether $A \cup B \cup C$ can be written as the disjoint union of n triangles from $\mathcal{T}$.

Suppose that we are given an instance of 3D-MATCHING as above. We show how to construct an instance of INTEGER RESILIENCE with $m + 3n + 2$ nodes, $11m + 3n$ arcs each of cost 1, n, or $2n$, such that there is a $(4m + 3n - 1)$-resilient integer reservation of cost at most $(2n + 1)(4m + 3n) + n$ if and only if the original instance did possess a 3D-matching.

We take one node of D for each triangle $abc \in \mathcal{T}$, one node for each element of $A \cup B \cup C$, and also nodes s and t. We take four parallel arcs, each of cost 1, from s to each node corresponding to an element of $\mathcal{T}$. Each node abc has seven arcs leaving it: four, of cost $2n$, go directly to t and one, of cost n, to each of the constituent elements a, b and c. Finally, there is a single arc of cost n from each element of $A \cup B \cup C$ to t.

Given a 3D-matching $\mathcal{U}$, we can find a $(T, 1)$-resilient reservation, with $T = 4m + 3n - 1$, by reserving capacity 1 on all arcs entering t, and all arcs leaving elements of $\mathcal{U}$; we further reserve capacity 1 on arcs from s to elements of $\mathcal{T} \setminus \mathcal{U}$, and capacity 2 on arcs from s to elements of $\mathcal{U}$. It is easy to check that this reservation is $(T, 1)$-resilient, and has the required cost $(2n + 1)(4m + 3n) + n$.

Conversely, if there is a $(T, 1)$-resilient reservation with this cost, one may easily check that it must involve reserving capacity 1 on all arcs into t, and also on one arc entering each node of $A \cup B \cup C$. Such a set of arcs has total cost $2n(4m + 3n)$, and includes between 4 and 7 arcs from each node of $\mathcal{T}$. For $v \in \mathcal{T}$, let $d(v)$ be the number of reserved arcs leaving v.

For $v \in \mathcal{T}$, the reservations on the four arcs between s and v must total at least $d(v)$, and the sum of any three of them must be at least $d(v) - 1$ (since after deleting an arc there still exists a T-flow). The minimum cost of such a reservation between s and v is just 4 if $d(v) = 4$ but $d(v) + 1$ if $d(v) \in \{5, 6, 7\}$. So the total cost of a $(T, 1)$-resilient reservation consistent with the values $d(v)$ is $2n(4m + 3n) + \sum_v d(v)$ plus the number N of elements v for which $d(v) > 4$. We know that $\sum_v d(v) = T + 1 = 4m + 3n$, and one may check that $N \geq n$, with equality if and only if $d(v) = 7$ for just n elements of $\mathcal{T}$, and $d(v) = 4$ for the remainder. The arcs out of $\mathcal{T}$ can be distributed in such a manner only if those elements v with $d(v) = 7$ constitute a 3D-matching in the original instance. □

On the positive side, there is a simple $(k+1)$-approximate algorithm for INTEGER RESILIENCE, namely to find a cheapest set of $k + 1$ edge-disjoint paths and reserve capacity T on each arc. The following result states that this is indeed a $(k + 1)$-approximate algorithm.

Proposition 7.2 *If x is a (T, k)-resilient reservation vector, then* $\text{cost}(x) \geq \frac{1}{k+1}\text{cost}(z^{k+1,k})$.

Proof. Let x be a minimum cost (T, k)-resilient vector. Define x' by setting, for each arc a, $x'_a = \min((k + 1)x_a, T)$; it follows that $x' \geq x$.

Let S be any (s, t)-set. We claim that $x'(\delta^+(S)) \geq (k + 1)T$. Suppose first that there is a set K of $k + 1$ arcs in $\delta^+(S)$ such that, for $a \in K$, $x_a \geq \frac{T}{k+1}$, and so $x'_a = T$. Then

$x'(\delta^+(S)) \geq \sum_{a \in K} x'_a = (k+1)T$.

If instead $x_a < \frac{T}{k+1}$ for all arcs $a \in \delta^+(S)$ except for those in a set K of k arcs. Then $x(\delta^+(S) - K) \geq T$, and so $x'(\delta^+(S) - K) = (k+1)x(\delta^+(S) - K) \geq (k+1)T$. This proves our claim.

Now, since $x'(\delta^+(S)) \geq (k+1)T$ for each (s,t)-set S, there exists an (s,t) flow x'' of value $(k+1)T$ such that $x'' \leq x'$, so in particular no arc has reserved capacity more than T; thus x'' is a (T,k)-resilient vector.

As we remarked earlier, the diverse path reservation $z^{k+1,k}$ is a minimum cost (s,t) flow of value $(k+1)T$, subject to an upper bound T on the flow through any given arc. It follows that $\text{cost}(z^{k+1,k}) \leq \text{cost}(x'') \leq \text{cost}(x') \leq (k+1)\text{cost}(x)$. □

Thus we have that the minimum cost diverse-path reservation has at most $k+1$ times the cost of an optimal (T,k)-resilient reservation. Of course, this is of greatest interest in the case $k=1$. The ratio $k+1$ between the two optimal costs is best possible, as can be seen by considering a network D with three nodes s, u and t, $k+1$ arcs of cost 1 from s to u, and many arcs of large cost c from u to t.

8 Conclusions

Future Directions

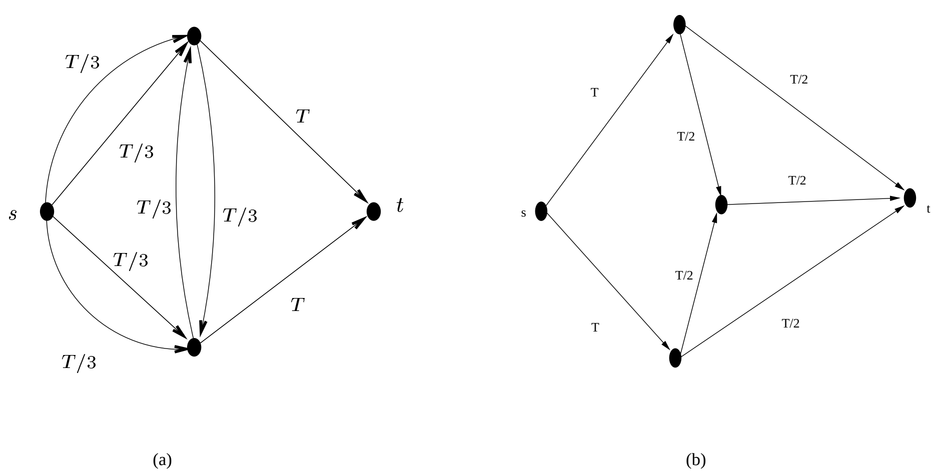

Figure 1: (a) vertex of $\mathcal{R}(T,1,D)$ with a cycle; (b) basic solution for both arc-failure and node-failure resilience.

Although General Resilience can be solved in polynomial time for fixed k, we have been unable to find a truly practical algorithm for the problem even in the case $k=1$. It is natural to believe that there might be an algorithm which uses some generalization of cycle augmentation for standard minimum cost flows. In order to explain why it is likely to be difficult to find such an algorithm, we give in Figures 1(a) and 1(b) two examples of vertices of polyhedra $\mathcal{R}(T,1,D)$, which of course give the unique optimal solution to instances of General Resilience. Indeed, the polyhedral structure for general resilience appears to be quite rich; further examples are provided in [11].

There are other versions of resilience questions which we did not investigate. For instance, one may wish to guard against node failures instead of or as well as arc failures: this problem can be formulated using our previous models and then applying standard splitting operations on nodes other than s and t. Figure 1(b) also represents a basic solution for such a fractional node-failure resilience problem.

In a further paper [12], we consider the effect of imposing upper bounds on the capacities that can be reserved on each arc.

Finally, another critical concern is how we may recover from failures. In the case of diverse path reservations, we have the best possible scenario. The node s can be programmed so that if a communication path fails, it simply shunts its traffic onto the remaining paths. For general reservation vectors, the rerouting of traffic may be more complex. Indeed, unless the reservation vector was carefully constructed, traffic on the non-failed communication paths may need also to be rerouted from scratch. The distinction between whether or not we may disturb non-failed traffic flows leads to two types (*strong* and *weak*) of resilience problems; these are discussed in some detail in the technical report [11].

Applications to More than one Source-Destination Pair

We now consider the problem where we are given a collection of node pairs $(s_1,t_1),\ldots,(s_q,t_q)$ as well as a collection of demands T_i, $i=1,2,\ldots,q$ and (possibly) a collection of integers $k_1,\ldots,k_q$. Each *commodity* i must reserve capacity in a network D which is (T_i,k_i)-resilient for the source-destination pair (s_i,t_i).

In developing solution techniques for a general multicommodity instance, we feel there is significant computational benefit in insisting that each commodity is handled by diverse-path reservation vectors. This is despite the fact that such reservations may cost more, even in the 1-commodity case (e.g., Figures 1(a) and 1(b)). This approach replaces the complexity of general resilience constraints with the simpler subproblem of deciding, for each commodity i, the number n_i of diverse paths to be included in the support of its reservation vector. This subproblem may be handled by some branching scheme and, for any fixed choice of the n_i's, the optimization problem can be formulated as a much simplified multicommodity network design problem.

For instance, suppose that for each arc a we may purchase up to M_a units of capacity, each at cost c_a. We then formulate the diverse-path multicommodity resilience problem, where each commodity i must use a diverse-path reservation on (exactly) n_i paths, as a mixed integer program.

$$\begin{array}{ll}
\min \sum_a c_a y_a & \\
\sum_{i=1}^q \frac{T_i}{n_i-k_i} r_a^i = y_a \le M_a & \text{for each arc } a \\
r^i(\delta^+(v)) = r^i(\delta^-(v)) & \text{for each } i \text{ and } v \neq s_i, t_i \\
r^i(\delta^+(s_i)) - r^i(\delta^-(s_i)) = n_i & \text{for each } i \\
r_a^i \in \{0,1\}, y_a \in \mathbb{Z} & \text{for each } a \text{ and } i
\end{array}$$

A solution r^i is thus an (s_i, t_i) 0-1 flow vector of value n_i, and the first family of constraints state that the total capacity reserved on an arc a is at most M_a.

Acknowledgements: The research of the first and second authors is supported by the EU-HCM grant TMRX-CT98 0202 DONET. They also acknowledge support from DIMACS during extended visits to Bell Labs. Some of the first author's research was also carried out while visiting the University of Memphis. The authors are grateful for insightful remarks and encouragement from Gautam Appa, Dan Bienstock, Fan Chung, Michele Conforti, Bharat Doshi, Susan Powell, Paul Seymour and Mihalis Yannakakis, as well as two anonymous referees.

A major inspiration for this work was Dr. Ewart Lowe, of British Telecom, who tragically died in a diving accident on May 22nd, 1998, off the coast of Normandy. Ewart introduced the authors to many mathematical problems in telecommunications. He also acted as mentor to the final author during his projects for British Telecom. We dedicate this paper to the memory of his inspiration, generosity, and his unbounded enthusiasm which are greatly missed by all who knew him.

References

[1] A.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Network Flows – Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[2] D. Alevras, M. Grötschel, R. Wessäly, Cost-efficient network synthesis from leased lines, *Annals of Operational Research* **76** (1998) 1–20.

[3] D. Alevras, M. Grötschel, R. Wessäly, Capacity and survivability models for telecommunication networks, *Konrad-Zuse-Zentrum für Informationstechnik Berlin, Technical Report* No. 97-24 (1997).

[4] N. Ascheuer, M. Grötschel, J. Rambau, Combinatorial online optimization in practice, *Optima* **57** March 1998.

[5] A. Balakrishnan, T. Magnanti, J. Sokol, Y. Wang, Modeling and solving the single facility line restoration problem, *Working Paper OR 327-98 Operations Research Center, MIT*, April 1998.

[6] F. Barahona, Network design using cut inequalities, *SIAM Journal of Optimization* **6** (1996) 823–837.

[7] J.J. Bartholdi, J.B. Orlin, H.D. Ratliff, Cyclic scheduling via integer programs with circular ones, *Oper. Res.* **28** (1980) 1074–1085.

[8] D. Bienstock, S. Chopra, O. Günlük, C. Tsai, Minimum cost capacity installation for multicommodity network flows, *Math. Programming B* **81** 2 (1998) 177–199.

[9] D. Bienstock, O. Günlük, Capacitated network design – polyhedral structure and computation, *INFORMS Journal on Computing* **8** (1996) 243–260.

[10] D. Bienstock, G. Muratore, Strong inequalities for capacitated survivable network design problems, *Math. Programming*, **89** (2000) 127–147.

[11] G. Brightwell, G. Oriolo, B. Shepherd, Some strategies for reserving resilient capacity, *LSE CDAM Report 98-04*, March 1998.

[12] G. Brightwell, G. Oriolo, B. Shepherd, Reserving resilient capacity with upper bound constraints, *LSE CDAM Report 2000-03*, March 2000.

[13] G. Brightwell, B. Shepherd, Consultancy report: Resilience strategy for a single source-destination pair, *LSE CDAM Report 96-22*, Aug. 1996.

[14] S.J. Bye, M. Herzberg, An optimal spare-capacity assignment model for survivable networks with hop limits, *Proc. GLOBECOM*, San Francisco, 1994, 1601–1606.

[15] S.J. Bye, M. Herzberg, Spare-capacity assignment in survivable networks for multi-link and node failures with hop limits, *Proc. Networks'94*, Sept. 1994, 381–386.

[16] S. Fortune, J. Hopcroft, J. Wyllie, The directed subgraph homeomorphism problem, *Theoretical Computer Science* **10** (1980) 111–121.

[17] A. Frank, Connectivity and network flows, in *Handbook of Combinatorics,* R.Graham, M. Grötschel, L. Lovász (eds.), North Holland Press, Amsterdam, 1992.

[18] M. Grötschel, C.L. Monma, M. Stoer, Design of survivable networks, in *Network Models, Handbooks in Operations Research and Management Science,* North Holland Press, Amsterdam, 1995.

[19] O. Günlük, Branch-and-cut algorithm for capacitated network design problems, (1996).

[20] R.R. Irashko, M.H. MacGregor, W.D. Grover, Optimal capacity placement for path restoration in mesh survivable networks, *IEEE International Conference on Communications*, Dallas, (1996) 1568–1574.

[21] T. Magnanti, P. Mirchandani, R. Vachani, Modelling and solving the capacitated network loading problem, *MIT OR Working Paper* OR 239-91 (1991).

[22] T. Magnanti, P. Mirchandani, R. Vachani, The convex hull of two core capacitated network design problems, *Math. Programming* **60** (1993) 233–250.

[23] T. Magnanti, P. Mirchandani, R. Vachani, Modelling and solving the two facility capacitated network loading problem, *Operations Research* **43** (1995) 142–157.

[24] T. Magnanti, Y. Wang, Polyhedral properties of the network restoration problem - with the convex hull of a special case *Working Paper OR 323-97, Operations Research Center, MIT*, November 1997.

[25] H. Sakauchi, Y. Nishimura, S. Hasegawa, A self-healing network with an economical spare-channel assignment, *Proc. GLOBECOM*, San Diego (1990) 438–443.

[26] I. Saniee, Optimal routing designs in self-healing communications networks, *Int. Trans. Op. Res.* **3** (2) (1996) 187-195.

[27] B.D. Venables, W.D. Grover, M.H. MacGregor, Two strategies for spare capacity placement in mesh restorable networks, *IEEE International Conference on Communications*, Geneva, (1993) 267–271.