Some Strategies for Reserving Resilient Capacity

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

Abstract

We examine various problems concerning the reservation of non-shared capacity for traffic in a network between source-destination pairs so as to be "resilient" against one or more network failures. In the simplest version of the problem we consider, the resilience requirement is that for a pair (s,t), on the failure of any one arc of the network, there is sufficient reserved capacity in the remainder of the network to support an (s,t) flow of some given value T. The most striking results come in 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 minimum cost network flows. Unlike traditional network flow problems, however, the integral version does not come for free: we do however give a polynomial time $\frac{1}{14}$-approximation algorithm, and show that this is best possible. These algorithms can also be adapted to solve other versions, for instance in an acyclic network when the reserved capacity has to constitute an integer flow. We leave many problems open, and provide a number of examples.

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

1 Introduction

1.1 Overview

This version of the paper includes, along with the technical details, many examples and discussion of some of the modelling complications. We begin with a discussion of the type of problem we wish to tackle, deferring precise definitions for the moment. For the majority of the paper, we consider traffic between a single pair of nodes; applications to the more general situation where there are many such pairs are discussed in Section 5.

We consider problems where we are required to reserve capacity in a network (directed graph) for traffic between a single source-destination pair (s, t) of nodes. 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. In a practical situation, one may imagine the network as being used to support traffic between other pairs of nodes, but the key point is that only the capacity reserved for (s, t) traffic may be used for that purpose. Our objective is to minimize the total cost of the reservations we make, subject to supporting a given target amount T of traffic from s to t (i.e., the reservation vector must dominate a network flow of T units from s to t).

In the absence of any other features, the problem described above is simply that of finding a minimum-cost path from s to t. However, if we do simply reserve capacity along a single path, we make ourselves totally vulnerable to the failure of any arc (or node) along our chosen path. For many purposes, this is not acceptable, and we wish to reserve our capacities so as to allow for the failure of any one arc of the network. (Most of what we do extends to settings where we require resilience against the failure of any k arcs, and/or where node-failures are possible, but for the moment we consider just the case of a single arc failure.)

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. We consider three types of constraints on the capacity reservation vector,

- *Integrality:* We may be forced to reserve capacity in discrete amounts, so that our reservation vector must be integral.
- *Type of resilience:* In the most basic version of the problem (*weak resilience*), all we require is that the reserved capacities must admit a T-flow from s to t on the failure of any one arc. However, we may additionally be required (*strong resilience*) to recover from the failure of an arc without disturbing any existing traffic not using that arc.

- *Structural:* For one reason or another, it may be required that our reservation vector be formed by selecting a collection of arc-disjoint (s,t) paths, and assigning a capacity to each path – we call such a reservation a *diverse-paths reservation.*[1] Alternatively, the capacities may be required to support (s,t) traffic of some given amount $M > T$ in the absence of any arc failures, thus we are seeking an upper bound on the percentage of lost traffic upon an arc failure.

One main result provides an extremely simple and practical combinatorial algorithm to find a minimum cost weakly resilient diverse-paths reservation. Examples of such reservations are obtained by selecting any $j \geq 2$ arc-disjoint paths, each with capacity $T/(j-1)$. We will see that an optimal solution will always be of this form. These solutions have other features that are attractive in practice: they are automatically strongly resilient, and the strategy for restoring the traffic in the event of an arc failure is extremely simple. If the paths that we are allowed to use are pre-specified and constitute a set of diverse paths, then the algorithm can be adapted to solve the integral version of the problem in strongly polynomial time.

In fact, this was the version of the problem which first sparked our interest, as it arose out of a practical problem encountered by British Telecom, and solved by two of the authors in a consultancy report [6] – this work forms the basis of Section 2. The research described in this paper is basically an attempt to generalize the results discovered in that case.

We shall give many examples to show that a solution of the general weak resilience problem need not look at all like a diverse-paths reservation. Our results for other versions of the problem are far from complete, although we do observe that the general weak resilience problem can be solved in polynomial time (although the integral version is NP-complete). One of the main problems posed is to develop a modification of the minimum cost flow algorithm to solve weak resilience. In practice, however, it might be reasonable to use the optimal diverse-paths reservation, even though it is not globally optimal: to support this view, we show that the best diverse-paths solution costs at most twice as much as the optimal weakly resilient solution, for any network.

Resilience problems are related to network reliability problems (see, for instance, Colbourn [9]), but seem to form a distinct area of study. There are evidently many versions of resilience, some of more practical value than others. The problems we tackle in this paper can essentially be viewed as capacitated extensions of problems of finding minimum cost networks with certain connectivity requirements. Excellent overviews of previous computational and theoretical work on such survivability and augmentation problems can be found in [13] and [11].

For the remainder of this section we introduce most of the notation required. We start by giving precise definitions, and specifying the formulations for resilience addressed in this paper. We then give a summary of our results.

[1] We will actually use the term diverse paths to refer to arc or node disjoint paths depending on the context.

1.2 Notation and Definitions

We give here the basic definitions and facts concerning flows in networks.

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 shall always assume that D comes with two nodes permanently fixed as the *source* (or *origin*) s and the *destination* (or *sink*) 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. Finally, we are also given a vector (c_a) of non-negative rational (again, usually integer) costs on the arcs a of D. Even though there may be parallel arcs, we often use the notation (u, v) when no confusion arises.

For any $S \subseteq V$, let $\delta^+_D(S)$ (or simply $\delta^+(S)$ if the context is clear) denote the set of arcs with tail in S and head in $V - S$. Let $\delta^-(S)$ denote the set of arcs $\delta^+(V - S)$. For a node $v \in V$, we write $\delta^+(v)$ for $\delta^+(\{v\})$, etc. If $S' \subseteq V$, we denote by $[S, S']$ the set of arcs with tail in S and head in S'.

We call $S \subseteq V$ an *(s,t)-set* if $s \in S$ and $t \in V - S$. An (s,t)-set S is called *s-minimal* if the graph induced by S contains a directed path from s to each node of S.

Let $\mathbb{Q}_+$ denote the set of non-negative rational numbers, so that $\mathbb{Q}^A_+$ is the set of all assignments of a non-negative rational to each member of the arc-set A, which we will frequently view as a vector. For any vector $x \in \mathbb{Q}^A_+$ and $A' \subseteq A$, we denote by $x(A')$ the sum $\sum_{a \in A'} x_a$. We let $I(x)$ denote the *support* of x, that is $I(x) = \{a \in A : x_a > 0\}$.

A vector $x \in \mathbb{Q}^A$ is an *(s,t) flow vector* (or simply a *flow*) if $x(\delta^+(v)) = x(\delta^-(v))$ for all $v \neq s, t$ and (therefore) $x(\delta^+(s)) - x(\delta^-(s)) = x(\delta^-(t)) - x(\delta^+(t))$. In this case, the *value* $val(x)$ of the flow is $x(\delta^+(s)) - x(\delta^-(s))$. The set of flow vectors of value M in the network D is denoted $\mathcal{F}(M, D)$.

For any rational M and (s,t)-set S, the *M-cut constraint* for S is:

$$x(\delta^+(S)) \geq M,$$

i.e., the total capacity of arcs leaving S is at least M. We define the M-cut polyhedron as

$$\mathcal{C}(M, D) = \left\{ \begin{array}{lll} x \in \mathbb{Q}^A : & x(\delta^+(S)) \geq M & \text{for each } (s,t)\text{-set } S \\ & x \geq 0 & \end{array} \right\}. \tag{1}$$

Network flow theory asserts that for a given vector $x \in \mathbb{Q}^A$, there is a flow vector $w \leq x$ of value M if and only if $x \in \mathcal{C}(M, D)$. It is an exercise to show that in (1) we need only include the cut constraints corresponding to s-minimal sets. Given such an x, we call a flow vector y *maximal* if $y \leq x$, and there is no flow vector g of greater value which satisfies $y \leq g \leq x$.

Consider now the *extended* digraph D^e obtained from D by adding $2(|V| - 2)$ new arcs as follows. For each $v \neq s,t$ create two arcs $r_v = (t,v), l_v = (v,s)$. Note that each vector $x \in \mathbb{Q}^A$, extends to a flow vector x^e as follows. For each $v \neq s,t$, define the *surplus* of node v as $s_v = x(\delta^+(v)) - x(\delta^-(v))$. For each $v \neq s,t$, if $s_v > 0$, then set $x^e_{r_v} = s_v$ and $x^e_{l_v} = 0$. Otherwise set $x^e_{l_v} = -s_v$ and set $x^e_{r_v} = 0$. For all other arcs a, set $x^e = x_a$. The vector x^e is called the *extension* of x and the *fractured flow value* of x, denoted by $\phi(x)$, is $val(x^e)$.

A *vertex* of a polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ is an extreme point of $\mathcal{P}$, i.e., a vector in $\mathcal{P}$ that is not a convex combination of two other vectors in $\mathcal{P}$. For a polyhedron $\mathcal{P}$ in $\mathbb{R}^n$, a vector $y \in \mathcal{P}$ is a vertex if and only if there is some linearly independent set of n inequalities defining $\mathcal{P}$ that are satisfied by y with equality. Since we consider only polyhedra in $\mathbb{R}^n_+$, if a linear function is bounded below on $\mathcal{P}$, then it attains its minimum at a vertex; such a vertex is called a *basic optimal solution* to the minimization problem.

1.3 Types of Resilience

We consider vectors $x \in \mathbb{Q}^A_+$ of reserved capacities on the arcs of D such that if an arc of D is deleted, then the remaining arcs have sufficient capacity. We consider three definitions for "sufficient".

WEAK RESILIENCE

A vector $x \in \mathbb{Q}^A$ of capacities is *weakly T-resilient* if, for each arc $a \in A$, the capacities on arcs in $A - a$ are sufficient to admit an (s,t) flow of value T. As in the previous section, this is equivalent to requiring that x satisfies the constraint $x(\delta^+(S) - a) \geq T$, for each (s,t)-set S and each arc $a \in \delta^+(S)$. In fact, it is easily shown that we need only require these constraints for s-minimal sets S. Such constraints are reminiscent of Seymour's characterization of the cycle cone for planar graphs [18].

As with the standard network flow problem, the problem of finding a minimum cost weakly T-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 arc $e \in \delta^+(S)$, the *partial T-cut constraint* associated with the pair (S,e) is:

$$x(\delta^+(S) - e) \geq T.$$

The *weak resilience polyhedron* is defined by the system of all partial cut constraints.

$$\mathcal{R}(T,D) = \left\{ \begin{array}{ll} x \in \mathbb{Q}^A : & x(\delta^+(S) - e) \geq T \quad \text{for each } (s,t)\text{-set } S \text{ and } e \in \delta^+(S) \\ & x \geq 0 \end{array} \right\}. \tag{2}$$

Note that $\mathcal{R}(T,D)$ is empty if there is an (s,t)-set S with $\delta^+(S)$ of size at most 1, and otherwise $\mathcal{R}(T,D)$ is full-dimensional. Also note that if $x \in \mathcal{R}(T,D)$, then setting any $x_e = 0$ results in a vector in $\mathcal{C}(T,D)$.

Thus if we reserve the capacities x, then even if an arc fails, there is still enough capacity to build an (s,t) flow of value T. Conversely, any vector x not in $\mathcal{R}(T,D)$ fails the partial T-cut constraint for some pair (S,e) and hence if e fails, there is not sufficient capacity in the network to support a flow of value T. Therefore $\mathcal{R}(T,D)$ consists exactly of the weakly T-resilient vectors. The problem of finding a mincost weakly T-resilient reservation is that of minimizing the linear function $\sum_{a\in A} c_a x_a$ over $\mathcal{R}(T,D)$. A natural way to approach the optimization problem in general is to investigate properties of the vertices of $\mathcal{R}(T,D)$, without explicit reference to a particular cost function.

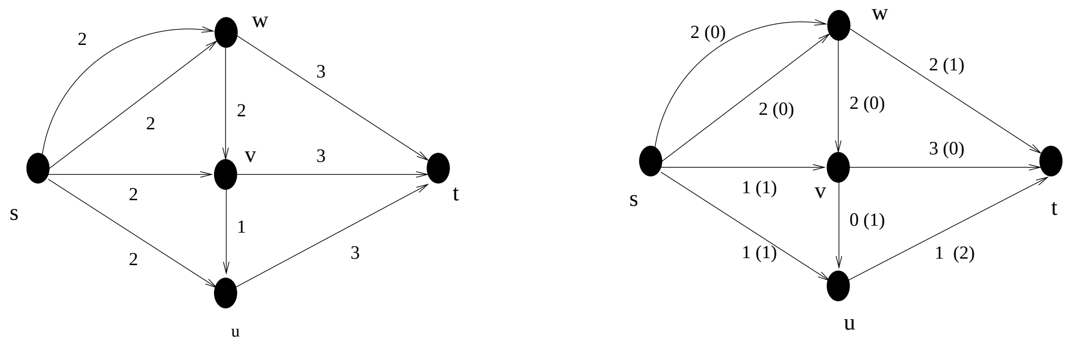
Another consequence of this formulation is that there is a polynomial-time algorithm to find a minimum cost T-resilient vector. Indeed, it is easily seen that the separation problem for $\mathcal{R}(T,D)$ is polynomially solvable. Still better, as noted by S. Powell and M. Yannakakis, the problem may be formulated as a linear program with a polynomially bounded number of variables and constraints. Namely, consider the set of vectors $(x, y^1, \ldots, y^m)$ such that each vector y^i is an (s,t) flow vector of size T with $y^i \leq x$ and $y^i_{a_i} = 0$, where $A = \{a_i\}_{i=1}^m$. Clearly the set $\mathcal{P}$ of such vectors can be expressed as the solution set to a system of constraints derived from the union of $|A|$ (s,t) flow defining linear systems. Moreover, $x \in \mathcal{R}(T,D)$ if and only if there exist y^i's such that $(x, y^1, \ldots, y^m) \in \mathcal{P}$. Thus we have a compact formulation for $\mathcal{R}(T,D)$ and so we may solve the minimization problem using any polynomial time algorithm for linear programming.

This is not offered as a practical algorithm however, and the remainder of the paper addresses the task of finding more direct combinatorial algorithms for solving these problems.

STRONG RESILIENCE

Weak resilience is the most natural and simple notion of resilience against an arc failure. However, we also wish to consider what happens in the case when the details of *how* we recover from an arc failure are important. We consider several scenarios where we view the reserved capacity as being sold to some customer who then may use it as they see fit. Thus at a given time we assume that there is a *working (s,t) flow* vector w which is established by the customer. Our concern is that the capacity allows for certain types of restoration of the working flow. A common and reasonable requirement is that, if there is existing (s,t) traffic using the network, which is not using the affected arc, then that traffic should not need to be redirected in the process of restoring the affected traffic. We focus on this scenario.

Let us see an example where this may be an issue. Consider the network in Figure 1. For appropriate costs, the arc values in Figure 1(a) form a (unique) minimum cost weakly 6-resilient capacity reservation vector. A "working flow" of value 6 for this network is displayed in Figure 1(b), where the numbers in brackets denote the "spare capacity" for this flow. Note however, that if the arc (u,t) were to fail, and we are asked to redirect the one unit of traffic currently flowing along the path $s \rightarrow u \rightarrow t$, then this cannot be done without altering flow values on paths which were unaffected by the arc's failure.



(a) (b)

Figure 1:

The crux of the problem with weak-resilience for this example is seen in the cut induced by the node set $S = \{s, u, v\}$. Let x be the reservation vector in Figure 1(a). Then $x(\delta^+(S) - (u,t)) = 7 > T$. However, the working flow uses all of these arcs – except (u,t) – at capacity, and furthermore involves a flow of 2 units along the "reverse arc" (w,v) into the set S. There is no way to recover from the failure of (u,t) without disturbing the traffic on the reverse arc.

The formulation of this problem turns out to be quite sensitive so let us now be precise about our requirements.

At a given time, there is a *working flow* w in the network which is actively carrying the traffic from s to t, with $w \leq x$, i.e., the working flow through any arc is at most its reserved capacity. The vector $x - w$ will then be the *spare capacity* at this time. We do not assume that the working flow vector even appears sensible - indeed it may contain cycles. This is because we assume that w may arise by a customer repeatedly establishing and then dropping "connections" between s and t. These connections evolve as demands for certain services arise. Any such connection will correspond to a "service" flow vector y which we assume contains no directed cycle in its support $I(y)$. Thus y is of the form $\sum_i \lambda_i \chi^{A(Q_i)}$ where the Q_i's are simple directed paths from s to t. A flow vector of this form is called *trail decomposable.* Each pair (Q_i, λ_i) is called a *trail.* Since w decomposes into these service flows, we also assume that w is trail decomposable: $w = \sum_{i=1}^{k} \mu_i \chi^{A(P_i)}$. We do not assume, however, that the support of w is acyclic.[2] Trail decomposability is enough to imply $val(w) = \sum_i \mu_i$ which may well be greater or less than T. Moreover, since a working flow w is trail decomposable, we see that for any arc a, w splits into two parts: a flow $f \leq w$ of value $val(w) - w_a$ comprising the traffic along paths P_i not using arc a, and a flow $w - f$ of value w_a, comprising the traffic using arc a. (This need not be the case if the P_i's are

[2] Clearly if $I(w)$ is acyclic, then w is trail decomposable. If $I(w)$ contains cycles, then w may or may not be trail decomposable.

allowed to be non-simple.)

We are now in a position to be more precise about how we may recover from an arc failure. Suppose that w is a trail decomposable working flow as above, and that a is in the support of w. After a fails, we wish to find a *restoration flow* r in $D-a$ such that $f+r \leq x$ and $f+r$ is a flow of value at least $\min\{val(w),T\}$. The critical question is what capacity is available for finding r? Two possible scenarios are:

1. Only the spare capacity $x-w$ is available for finding the restoration vector.
2. We may use the spare capacity $x-w$ and in addition, the capacity used by the disturbed flow $w-f$ is relinquished upon a's failure and hence can be used for restoration. Thus we search for a restoration flow in $D-a$ with capacities $x-f$.

Scenario **1** is perhaps less realistic; it also has the simpler mathematical formulation. In this case, one checks that the reservation vector x must satisfy the condition **c.1**: *for each maximal flow vector f, and for each arc a, we have $f_a \leq val(f)-T$. In particular, any maximal flow has value at least T.*

Scenario **2** is complicated but manageable. Let w be a working flow and a an arc in the support of w. If a fails, then we must restore w to a new flow vector of value at least $\min\{T, val(w)\}$. This should be done without disturbing the trails which do not use the arc a. So suppose w is decomposed into f and $w-f$ as above. Our resilience condition becomes: **c.2**: *For any w,f,a as above, the network $D-a$ with capacities $x-f$ contains a flow vector of value at least $w_a - \max\{0, val(w)-T\}$. In addition, $x \in \mathcal{C}(T,D)$.*

Since we are examining the case where our reserved capacity x must be able to recover from a failure for *any* working flow found, it is reasonable to ask whether x should not also be robust in the process of finding the working flows. In particular, should we guarantee that no matter how such a flow is found, it will achieve at least the target failure rate T? That is we may also require the extra condition **c.3**: *the reservation vector x must have the property that any maximal working flow has value at least T.* Note that this is automatically satisfied if x satisfies **c.1**.

Actually, if we impose conditions **c.1** and **c.2**, then one can deduce that our task is described purely in terms of the flow f that we are not allowed to disturb. We choose to work with essentially this formulation, and accordingly we call a vector $x \in \mathbb{Q}^A_+$ *strongly T-resilient* if for each subgraph $D-a$ and each working flow vector $f \leq x$ in $D-a$, there exists a flow vector g in $D-a$, of value at least T such that $f \leq g \leq x$.

Strong resilience is an unpleasant property in many ways. It is not in general *monotone*: there are networks with reservation vectors x and y such that $x \leq y$, x is strongly T-resilient and y is not. Perhaps even worse, it is not in general *convex*: there are networks with strongly T-resilient reservation vectors x and y such that $\frac{1}{2}x + \frac{1}{2}y$ is not strongly T-resilient. [3] In fact, the problem of recognizing whether a vector is strongly resilient is NP-hard - see Section 4. Nevertheless, in some situations the practical problem is to find such strongly resilient reservations. We offer no suggestions for how to solve this problem in general, but it seems to us to be worthy of further study.

We do however consider conditions that are sufficient for strong resilience, but apparently somewhat more tractable. For instance, it is easy to imagine a setting in which our strongly T-resilient vector is itself required to be a minimum cost working flow of some value $M > T$. It turns out that we can give a simple algorithm for solving the problem under these conditions - see Section 3.

We consider a last restoration scenario which does not submit to our formulations. If speed to restoration is important, then each of the trails (P_i, μ_i) using a failed arc, may be restored separately and in restoring P_1, say, capacity on the non-failed arcs of this path are still available for the restoration of this trail. The capacities reserved by the other trails are not available. (Spare capacity is always available.)

NET RESILIENCE

As we saw with the example in Figure 1, the extra constraint of strong resilience requires us to pay attention to reverse arcs going into s-minimal sets, whereas the formulation for weak resilience allows us to ignore these. However, the following seems a natural way of dealing with reverse arcs: as we shall see, it is actually a somewhat stronger condition than that of strong resilience.

The *net partial T-cut constraint* associated with a pair (S, e) is:

$$x(\delta^+(S) - e) - x(\delta^-(S)) \geq T.$$

A vector $x \in \mathbb{Q}_+^A$ is *net T-resilient* if it satisfies all the net partial T-cut constraints, in other words if it is in the *net resilience polyhedron*:

$$\mathcal{N}(T, D) = \left\{ \begin{array}{ll} x \in \mathbb{Q}^A : & x(\delta^+(S) - e) - x(\delta^-(S)) \geq T \quad \text{for each } (s,t)\text{-set } S \text{ and } e \in \delta^+(S) \\ & x \geq 0 \end{array} \right\}. \quad (3)$$

Again this polyhedron is non-empty as long as there is no (s,t)-set S with $|\delta^+(S)| \leq 1$. This polyhedral formulation shows that net resilience is a convex property. It is not monotone however: for instance, adding an arc with large enough capacity from t to s will always destroy net resilience.

As for weak resilience, the problem of finding a minimum cost net T-resilient reservation is thus that of minimizing a linear function over a well-described polyhedron. This may be solved in polynomial (in $|A|$ and $|V|$) time by any polynomial algorithm for linear programming - see Section 4 for the

[3] A key difficulty is coping with any possible working flow which may arise, for example, in a noncooperative network where customers may choose their own end to end connections. One moral to this has been previously observed, namely that the cost-effectiveness of the overall network suffers in such situations.

formulation. This is a positive feature, although net resilience in general proves to be too strong a condition. Nonetheless it is a useful stepping stone to understanding the concepts of weak and strong resilience. We begin with the good news.

Proposition 1 *Every net T-resilient vector is strongly T-resilient.*

Proof. Let $x \in \mathcal{N}(T, D)$, let a be any arc in D, and let $f \leq x$ be any flow between s and t in $D - a$, of value T_0 say. We need to find a flow of value $T - T_0$ in the network $D - a$ with capacities given by $y = x - f$.

There is such a flow, as long as $y(\delta^+_{D-a}(S)) \geq T - T_0$ for all (s,t)-sets $S \subseteq V$. Take any (s,t)-set S, and consider the flow f in relation to S. Since f is a flow of value T_0, we have that $f(\delta^+_{D-a}(S)) - f(\delta^-_{D-a}(S)) = T_0$. Also, since x is net T-resilient, we have $x(\delta^+_{D-a}(S)) \geq T + x(\delta^-_{D-a}(S)) \geq T + f(\delta^-_{D-a}(S))$. Thus

$$\begin{array}{rcl} y(\delta^+_{D-a}(S)) & = & x(\delta^+_{D-a}(S)) - f(\delta^+_{D-a}(S)) \\ & = & x(\delta^+_{D-a}(S)) - T_0 - f(\delta^-_{D-a}(S)) \\ & \geq & T - T_0, \end{array}$$

as required. □

At first glance, it might appear that net resilience is practically the same concept as strong resilience. However, this is far from the case, as the next example shows. We construct a network with node set $\{v_0, v_1, v_2, \ldots, v_{2k}, v_{2k+1}\}$, with $s = v_0$ and $t = v_{2k+1}$, for some k, with three parallel arcs from v_{2i} to v_{2i+1} $(i = 0, \ldots, k)$ and two parallel arcs from v_{2i-1} to v_{2i} $(i = 1, \ldots, k)$ – see Figure 2. If all arcs have the same cost, it is easy to check that the optimal weakly T-resilient reservation x is as shown in the Figure, and that x is strongly T-resilient also. However, it is quite far from net T-resilient. To see this, let $S = \{v_0, v_2, v_4, \ldots, v_{2k}\}$, the set of even-numbered nodes. Then $x(\delta^+(S)) - x(\delta^-(S)) = (k+1)\frac{3T}{2} - k2T = \frac{(3-k)T}{2}$. Thus for k large enough and any arc e whatsoever, the net partial T-cut constraint for (S, e) will be violated.

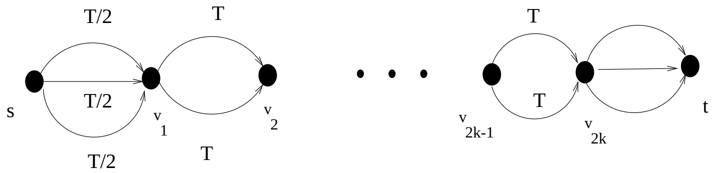


Figure 2: *A weakly and strongly resilient solution.*

The main feature of this example is that the reservation x is very far from being a flow. In the general situation, setting S equal to the set of nodes v with a surplus of in-flow, i.e. $x(\delta^-(v)) >$

$x(\delta^+(v))$, is always likely to strain the net partial cut constraints. Indeed, in order to check whether $x(\delta^+(S')) - x(\delta^-(S')) \geq T$ for each (s,t)-set S', one need only check this set. Hence to test membership in $\mathcal{N}(T,D)$, one does this for the appropriate set in each arc-deleted subgraph. Of course, in the example above, the set S is a senseless (s,t)-set to consider for restoring flows $f \leq x$, and indeed the reservation vector is strongly resilient and as well-behaved as one could wish for. In reality, we should not be imposing such a partial cut constraint on x unless the set S is s-minimal. Restricting to s-minimal sets gains little however, since one could include every arc not in the digraph of Figure 2, with cost ∞ say; S would then be s-minimal for the resulting digraph.

This example shows how strong (and strained) the property of net resilience is. On the other hand, it is the only tractable formulation we know which guarantees strong resilience. In fact, perhaps its main use is as a condition which guarantees that any greedily constructed working flow, will reach a value of T - condition **c.1** from the previous section. (One easily amends the proof of Proposition 1 to show that if x is net T-resilient, then every maximal flow has value at least T.)

One of the key differences between weak resilience and net resilience is that, given a directed cycle in the support of some resilient vector x:

- for weak resilience we may always add ϵ along the arcs of this cycle and remain resilient (but not necessarily subtract ϵ) – indeed we can add capacity on any arcs of D and remain weakly resilient
- for net resilience we may always subtract some ϵ along the arcs of this cycle and remain resilient (but not necessarily add ϵ).

Lemma 2 *If x is net T-resilient and C is a directed cycle in the support of x, then for sufficiently small $\epsilon > 0$, $x' = x - \epsilon\chi^{A(C)}$ is net T-resilient.*

Proof. Choose ϵ small enough for x' to be non-negative. Consider the net partial T-cut constraint for any (S,e); if this is satisfied by x, then it will also be satisfied by x', since the net flow out of S is unaffected, while the impact of deleting any one arc in $\delta^+(S)$ is not heightened. □

We say a reservation vector $x \in \mathbb{Q}^A_+$ is *acyclic* if there are no cycles in the support $I(x)$ of x. The previous lemma shows immediately that in optimizing over $\mathcal{N}(T,D)$ with nonnegative costs, we may assume that the solutions are acyclic reservation vectors. This is not the case for weak resilience - e.g., see Figure 3, showing a vertex of $\mathcal{R}(T,D)$ for the given digraph. For strong resilience, we do not know the answer to this question!

OTHER FORMS OF RESILIENCE

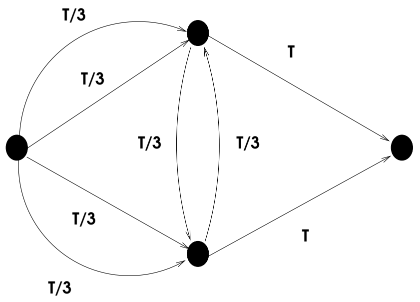


Figure 3: *A basic weakly resilient vector with a cycle.*

We have concentrated on the case where we require resilience against a single arc failure. Our methods extend to the case of resilience against multiple arc failures; we shall discuss this as we go along. This is however a good point to introduce the idea of resilience against node failures.

Given a digraph $D = (V, A)$, we define a reservation vector $x \in \mathbb{Q}_+^A$ to be (weakly) T-node-resilient if, for every node v, there is a T-flow f with $f \leq x$ whose support includes no arcs to or from v. We can model node resilience as a standard weak resilience problem using the following standard technique (splitting nodes).

Define an auxiliary digraph D' by replacing every node v of D (except s and t) with a pair of nodes v^-, v^+, with an arc from v^- to v^+ of cost 0. Each arc (u, v) in D is replaced by an arc (u^+, v^-) in D', with the same cost as in D. Reservations in D' essentially correspond to reservations in D, since we may as well purchase infinite capacity on each arc (v^-, v^+) in D'.

Proposition 3 *A reservation in D is T-node-resilient if and only if the corresponding reservation in D' is weakly T-resilient.*

One could also define notions of strong or net node-resilience, but we do not investigate these further.

Another possibility is that we may wish to guard against losing some fraction $\alpha \in [0, 1]$ of the flow on each arc. For weak resilience, for example, the appropriate constraints would be

$$x(\delta^+(S)) - \alpha x_e \geq T. \tag{4}$$

A vector x satisfying these constraints will be called $(\alpha; T)$-resilient. Note that if $\alpha = 0$, this leads to a formulation for traditional network flows. If $\alpha = 1$ it is the partial T-cut constraint again. The effect of intermediate values of α can essentially be analyzed by duplicating arcs.

To be precise, suppose that $\alpha = p/q$ is rational, and that we require an $(\alpha; T)$-resilient reservation x in a network D. To model this, we form D' by taking q copies of each arc of D, each with the same cost as in D. It is easy to check that a minimum cost reservation vector for D' that is weakly qT-resilient against the loss of any p arcs can be constructed from a minimum cost $(\alpha; T)$-resilient reservation in D by duplicating the reservation on each set of multiple arcs.

The final type of resilience we consider arises if the structure of the reservation vector x when all arcs are functional is of paramount importance. A likely scenario is that there is some value M such that the vector x is required to admit an M-flow, that is, there is an (s,t) flow f of value M with $f \leq x$. We don't discuss this constraint, but we do discuss the variation where the vector x is further required to itself be a flow of value M. This corresponds to a situation where we are seeking a working flow for traffic when the network is operating normally, but where the capacities must support a flow of value T on the failure of any one arc. Alternatively, this can be thought of as being a constraint on the percentage loss of service on the failure of an arc.

Accordingly, we call a vector $x \in \mathbb{Q}_+^A$ *weakly/strongly/net* (T,M)*-resilient* if it is weakly/strongly/net T-resilient, and in addition lies in the polyhedron $\mathcal{F}(M,D)$, i.e., it is a flow of value M.

1.4 An Overview of the Results

Section 2: Resilient Diverse-Paths Reservations

In this section, the notions of strong, weak and net resilience are equivalent. In fact, due to the simplicity of the cases studied we can model the problems attacked without referring to the LP definitions introduced above.

We start with the case where we are given a network *and* a collection $P_1, P_2, \ldots, P_m$ of pre-selected arc-disjoint paths from s to t. These paths have associated costs $c_1 \leq c_2 \leq \ldots \leq c_m$, which measure the expense of reserving a unit of capacity along the entire path.

We solve the problem of finding a minimum cost weakly/strongly/net T-resilient reservation of the form $\sum_{i=1}^m x_i \chi^{A(P_i)}$. We note that this optimal reservation is always of the form $\sum_{i=1}^j \frac{T}{j-1} \chi^{A(P_i)}$, for some $j \geq 2$.

We remark that any derived solution for this problem will also guard against multiple failures on a single path at no extra cost (due to the arc-disjoint nature of the paths). If the paths also contain nodes whose failure is to be allowed for, then the results of this section apply, provided the paths are internally node-disjoint.

We also solve the variation of the problem where traffic is to be guarded against the simultaneous failure of $k \geq 1$ arcs; the solution is similar to that of the 1-failure case.

For both problems, we describe a very simple method for determining the optimal allocation of capacities. We also give a solution (and polynomial-time algorithm) for the related problem where the reserved capacities x_i must themselves be integers. This condition arises, for instance, in telephone networks where the reserved capacities correspond to bandwidths that need to be allocated in bundles (e.g., STM-16 trails).

In the above problems, the candidate paths P_i are given at the outset. It may then be that our assignment of capacities is optimal for this collection of paths, but that there exists a collection of paths providing a cheaper solution. We also consider the problem of selecting the paths as part of the optimization process. We give a simple algorithm to find a minimum cost weakly/strongly/net resilient diverse-paths reservation in any network. The integer version of this problem turns out to be NP-hard. In this case we give a polytime $\frac{1}{14}$-approximate algorithm, and show this to be the best possible.

Section 3: Flow-Reservations and (T, M)-Resilience

In this section we drop the constraint that the reservation vector must be a diverse-paths reservation, but we do impose the weaker condition that it be a flow vector. This is the case, for instance, when the reservation vector must itself form an initial working flow. In this instance, there is a dramatic difference between strong or net resilience and weak resilience.

It turns out that the optimal net resilient reservation will again always be a diverse-paths reservation. We also conjecture that the net and strong resilience problems coincide for flow reservations, and show that this is the case under certain conditions.

Partly as a tool, and partly for its own sake, we use the concept of (T, M)-resilience here. We show that net resilience and strong resilience coincide for acyclic flow vectors, and explain how to use the algorithms of the previous section to find a minimum cost strongly (T, M)-resilient reservation. Among other things, this adapted algorithm can be used to find a minimum cost flow-reservation with integer capacities.

It is shown that the weak resilience problem restricted to flow vectors is in general no easier than the full weak resilience problem (and so a result in Section 4 will show that the integral version is NP-hard). On the other hand, in the special case where the network is acyclic, the methods for net resilience also apply for weak and strong resilience.

Section 4: Resilient Reservations with no Restrictions

In this section, we discuss the general problems of weak, strong and net resilience with no side-constraints. We show that the problem of recognizing a strong resilient reservation is NP-hard. We also give several examples of possible optimal solutions, showing that we do not always obtain anything resembling a diverse-paths reservation in the general case, for any of the three forms of resilience.

We give here a polynomially bounded linear programming formulation for the net-resilience problem. As already noted, minimum cost weak (or net) resilient vectors may also be found in polynomial time by linear programming algorithms. In this section we show, however, that the integral version of the weak, strong and net resilience problems are all NP-hard, and in the case of weak resilience, even finding minimum cost integer flow-reservations is hard.

However, we do show that the naive algorithm that allocates capacity T to the arcs of the cheapest pair of arc-disjoint (s,t) paths is a 1-approximation algorithm to any of the three problems. In other words, the optimal solution is always at least half the cost of this solution. We offer this as support for the approach of using the optimal diverse-paths solution as a good heuristic for the general problem.

Section 5: Applications to More than one Source-Destination Pair

We close by examining the problem of finding resilient capacity allocations where more than one pair of nodes is offering traffic to the network. This problem and its relatives has been addressed by a number of groups from a computational viewpoint, e.g., [5], [3], [4], [19], [7], [8], [15], [16], to name just a few. These often involve very complicated linear integer programs which look like a conglomeration of $|A|$ network loading mixed integer programs. We examine more closely the simple approach based on allowing only diverse-paths reservations.

2 Resilient Diverse-Paths Reservations

As we have mentioned, our aim with resilience problems is often to reduce to the case where the reservation has to form a collection of arc-disjoint paths, and we term such a reservation a *diverse-paths reservation*. (We use this term so as to include the case where node failures are allowed, and the paths are internally node-disjoint.) We thus need to consider how we solve the problem once it has been reduced to this special case.

We start with the case where the collection of diverse paths is fixed ahead of time. This forms the basis for much of what follows and so we present it in detail. The material in the first two parts of this section, written for a non-technical audience, appears as [6]. A more thorough handling of the polyhedron considered herein (including a complete linear description of the integer hull) has been given by Bienstock and Muratore - see [5].

2.1 Reservations on a Set of Fixed Paths

Suppose that we are given a network, together with a source s and a destination t, along with a fixed set of diverse paths $P_1, \ldots, P_m$ from s to t on which capacity may be reserved for (s,t) traffic. We want to ensure that, in the absence of any one path P_i, there is sufficient capacity on the remaining paths to carry a given amount T of traffic from s to t. To accomplish this requires us to fix a capacity x_i for each path P_i, and give each arc on path P_i capacity x_i. The cost of reservations can be calculated from the per-unit total costs c_i of the paths P_i. Thus in practice we can think of the network as consisting of just the two nodes s and t, with the P_i being single arcs of cost c_i from s to t.

A diverse-paths reservation as above is automatically weakly, strongly and net resilient. If the paths are internally node-disjoint, it is also node-resilient. We may assume that the paths P_i are numbered in increasing order of per-unit cost c_i, so we can state the problem formally as follows.

Single-Failure Allocation Problem.
Given a demand T, and a sequence of costs $c_1, c_2, \ldots, c_m$, with

$$0 \leq c_1 \leq c_2 \leq \ldots \leq c_m, \tag{5}$$

find non-negative real numbers $x_1, x_2, \ldots, x_m$ to minimize $c_1x_1 + c_2x_2 + \ldots + c_mx_m$ subject to the conditions

$$\text{For any } l, \quad \textstyle\sum_{i \neq l} x_i \geq T. \tag{6}$$

We shall also consider the integral version of this problem, where the x_i are constrained to be integers, and the k-failure version of the problem.

We first present the solution to the Single-Failure Allocation Problem. We give an example, and describe how the solution generalizes to the variations mentioned above. We defer the proofs until Section 2.2.

SOLUTION TO THE SINGLE-FAILURE ALLOCATION PROBLEM

A result that is fundamental in much of the rest of the paper is that the optimal allocation of capacities (for any costs satisfying (5)) is always a vector z^j of the form

$$z^j : \quad z_i^j = \begin{cases} T/(j-1) & i \leq j \\ 0 & i > j, \end{cases}$$

for some value of j between 2 and m. In other words, only the cheapest j paths are used at all, and those will all have been allocated the same capacity $T/(j-1)$. Note first that z^j is indeed a solution to the problem, since the loss of any one path leaves at least $j-1$ active paths with total capacity T, as needed. Second, note that the cost incurred by solution z^j is simply

$$\sum_{i \leq j} c_i \frac{T}{j-1}. \tag{7}$$

Choosing amongst the various solutions z^j cannot be hard, but in fact the structure of the problem allows us to simplify even this, as we now describe. For $j = 2, \ldots, m$, let A_j denote the average (cost) of the $j-1$ values $c_1 + c_2, c_3, \ldots, c_j$. From (7) we have that the cost of solution z^j is $A_j T$; so we are simply trying to find the smallest such average A_j.

To do this, first compare $A_2 = c_1 + c_2$ with c_3; if A_2 is smaller, then z^2 is optimal – if not, then A_3 is smaller than A_2 (and hence z^3 is better than z^2). In the latter case, we continue and compare $A_3 = (c_1 + c_2 + c_3)/2$ with c_4; if A_3 is smaller, then z^3 is optimal – if not then as before, z^4 is better and we must continue. We proceed until we find the optimal solution. If we reach the final possibility z^m and do not rule it out, then it is optimal. It is a simple exercise to show that this procedure does correctly identify the least A_j and therefore the optimal solution to the problem.

Example

There are five paths, with costs $c_1 = 3$, $c_2 = 4$, $c_3 = 6$, $c_4 = 8$, $c_5 = 8$, and the demand is 60. The four potential solutions are:

z^1: Capacity 60 on each of the first two paths.

z^2: Capacity 30 on each of the first three paths.

z^3: Capacity 20 on each of the first four paths.

z^4: Capacity 15 on each of the paths.

To find which is best, we could just compare costs, but it is more efficient to use the above method, as follows.

(a) Compare $c_1 + c_2$ with c_3; c_3 is smaller, so z^2 is better than z^1.

(b) Compare $(c_1 + c_2 + c_3)/2$ with c_4; c_4 is larger, so z^2 is optimal.

Note that z^2 would still be optimal even if there were more paths (with costs at least 8) beyond the fifth one.

THE k-FAILURE ALLOCATION PROBLEM

We now consider the case where we want any $m-k$ of the m paths to have combined capacity T. That is, if there are k path failures, the surviving paths have enough reserved capacity to carry the traffic.

The optimal solution for this will always be of the form:

$$z^{j,k}: \quad z_i^{j,k} = \begin{cases} T/(j-k) & i \le j \\ 0 & i > j, \end{cases}$$

for some j between $k+1$ and m. As before, we need only find the appropriate j for which the cost is minimized. Also essentially as before, the best candidate is obtained by choosing j to minimize the average of the $j-k$ costs $c_1 + \ldots + c_{k+1}, c_{k+2}, \ldots, c_j$. This minimum can be found in exactly the same way as for the basic case.

Of course, the single-failure problem is just the special case of the k-failure problem with $k = 1$.

ALLOCATION PROBLEMS WITH INTEGER CAPACITY ALLOCATIONS

To find the optimal solution in the case where the allocated capacities x_i are required to be integers (and the demand T is an integer), the following procedure suffices.

First find the optimal solution $z^{j,k}$ to the original (non-integral) k-failure Allocation Problem. Then (if the $z_i^{j,k}$ are not already integers) consider the two integer solutions "nearest" to $z^{j,k}$, as follows.

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

(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 $l = \lceil T/r \rceil + 1$; if $l \le m$, set $x_1 = x_2 = \ldots = x_{l-1} = r$, $x_l = T - (l-k-1)r$, and $x_{l+1} = \ldots = x_m = 0$. (The choice of l ensures that $x_l \le x_1 = r$. If $r = \lfloor T/(j-k) \rfloor$, we could have $l > m$, 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 $(l-k-1)r + x_l$, 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). To finish, just calculate the costs of the two solutions, and choose the lower.

2.2 The Proofs

This section is devoted to the proofs of the results discussed in the previous section.

Theorem 4 (i) *The optimal solution to the Single Failure Allocation Problem is obtained at one of the solutions* z^j.
(ii) *The optimal solution to the* k*-Failure Allocation Problem is obtained at one of the solutions* $z^{j,k}$.

Proof. We just prove (ii), since (i) is a special case.

Because of the symmetry of the situation, we know that, at the optimum solution, we will have $x_1 \geq x_2 \geq \ldots \geq x_m$. 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* $c_1 < c_2 < \ldots < c_m$*, 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.

This reformulation is in the form of a Linear Program, so the optimum is attained at a vertex, where a linearly independent set of m of the inequalities are satisfied with equality. If there are j non-zero variables at the optimum, 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 $z^{j,k}$.

The conclusion is that the optimum is attained at one of the solutions $z^{j,k}$, as claimed. □

CHOOSING AMONG THE SOLUTIONS $z^{j,k}$

The cost of solution $z^{j,k}$ is $TA_{j,k}$, where $A_{j,k} = (c_1 + \ldots + c_j)/(j-k)$, which is best thought of as the average of $c_1 + c_2 + \ldots + c_{k+1}$, c_{k+2}, $\ldots$, and c_j. We are trying to minimize $A_{j,k}$, over the range of possible values $j = 2, \ldots, j = m$. It is simple to check that the procedure described in the previous section for the single-failure case solves the general k-failure problem too. We note for future reference that the $A_{j,k}$ are in fact decreasing in j up to the minimum value, and increasing thereafter.

INTEGER SOLUTIONS

Again, we work in the general k-failure case; the basic integral problem is then the case $k = 1$.

Theorem 5 *Suppose we have an instance of the k-Failure Allocation Problem in which the optimal solution is $z^{j,k}$. Then the optimal solution x with all the x_i integer 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, which, as we noted, is also valid for the integer case. Suppose that x is an optimal integer solution.

Clearly, since the only constraints on the first k variables are that $x_1 \geq x_2 \geq \ldots \geq x_{k+1}$, we will 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 $k+2 \leq i < j \leq l$. Consider increasing x_i by one, and decreasing x_l by one. This keeps the solution feasible, since all the variables remain in the right order, and $x_{k+1} + \ldots + x_m$ is unaltered. Also, this operation decreases the cost.

We have thus shown that one may restrict attention to integral solutions with the following form: for some j with $k+2 \leq j \leq m$, we have all of $x_1, \ldots, x_{j-1}$ equal, and all of $x_{j+1}, \ldots, x_m$ 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 will be an integer, at least q.

At this point, there are still potentially a large number of candidate solutions; in fact, there is one for each integer value of x_1 at least T/m, namely to set $j = k + \lceil T/x_1 \rceil$, and $q = T - (j-k-1)x_1$.

We next observe that the integer solution given above is a convex combination of the two "fractional" solutions $z^{j,k}$ and $z^{j-1,k}$. To be precise, our solution is

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

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 $z^{k+1,k}, \ldots, z^{m,k}$. We note that any vector on this path gives a feasible solution. Next observe that the first coordinate values are decreasing along $\mathcal{A}$, and the values for any vector in $\mathcal{A}$ are thus determined by the first coordinate x_1. Moreover, this solution will be integral if and only if x_1 is an integer. Also, the argument for the fractional case (and convexity) shows that the corresponding solution cost is a unimodal function along $\mathcal{A}$, with the minimum obtained at the vertex corresponding to some $z^{j,k}$, with first coordinate equal to, say, r. These arguments show that the lowest cost integral solution among our candidates, and so the overall optimal integer solution, is the solution on $\mathcal{A}$ corresponding to taking x_1 to be either $\lceil r \rceil$ or $\lfloor r \rfloor$, i.e., taking either $z^{j,k,+}$ or $z^{j,k,-}$. This completes the proof. $\square$

We close this section with a bound relating the costs of the optimal integral and fractional solutions.

Proposition 6 *For any $j > k \geq 1$, let O be the cost of a solution $z^{j,k}$ and O' be the cost of the integral solution $z^{j,k,-}$. Then $O'/O < 1 + \frac{k(j-k)}{jT} < 1 + \frac{k}{T}$.*

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 $l-1$ cheapest paths with capacity $x = \lceil T/(j-k) \rceil$, and the next cheapest path with capacity $T-(l-k-1)x$, where $l = \lceil T/x \rceil + k \leq j$.

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 O'/O 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 $(l-1)x + T - (l-k-1)x = T + kx$. Hence we have

$$\frac{O'}{O} \leq \frac{T+kx}{jT/(j-k)} = 1 + \frac{k}{jT}\left((j-k)x - T\right).$$

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

Corollary 7 *If O_I is the cost of the optimal integral solution to an instance of the k-failure allocation problem with target flow T, and O_F the cost of the optimal fractional solution to the same problem, then $O_I < (1 + k/T)O_F$.*

Consideration of the proof of Proposition 6 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 $M \to \infty$, $O_I/O_F \to (1 + k/T)$.

2.3 Diverse-Paths Reservations without Specified Paths

We now consider what happens 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.

In this case, the results of Section 2.1 imply that the optimal solution will have as support the arcs of some j diverse paths, with each arc in the support given capacity $T/(j-1)$ (or, more generally, $T/(j-k)$ in the k-failure case). Of course, j can only take integer values up to the (s,t)-connectivity $\kappa = \kappa(D)$ of D. There is an efficient algorithm (described below) to find the minimum cost collection of j arc-disjoint paths from s to t in D, so the optimal solution can be found by simply running this algorithm for each j, and adopting the cheapest of the solutions found.

Closer examination of the j-shortest-paths algorithm yields a slight simplification of this procedure, and the features of this algorithm will turn out to be important later, so we need to discuss it in detail.

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 algorithm, we construct a series of auxiliary digraphs D_j, each of which will contain exactly one of each pair a, a^-.

Proposition 8 (cf. [17]) *Let c be a nonnegative vector of arc costs in a network $D = (V, A)$. The following algorithm finds a minimum cost collection of j arc-disjoint (s,t) paths in D, with arc-set $\mathcal{P}_j$, for each j from 1 up to $\kappa(D)$.*

Paths(D)
{
    $j = 0; D_0 = D; c^0 = c; \mathcal{P}_0 = \emptyset;$
    `While` (D_j `contains a directed` (s,t) `path`)
        `Let` Q_j `be a minimum` c^j`-cost directed` (s,t) `path in` D_j
        `Set` $\mathcal{P}_{j+1} = (\mathcal{P}_j - R) \cup F$
            `where` $R = \{a \in A : a^- \in E(Q_j)\}$
            `and` $F = A \cap E(Q_j)$
        D_{j+1}, c^{j+1} `are the same as` D_j, c^j `except`
            `if` $a \in R$
                `remove` a^-`, and include` a `with cost` $c_a^{j+1} = c_a$
            `if` $a \in F$
                `remove` a`, and include` a^- `with cost` $c_{a^-}^{j+1} = -c_a$
        `Set` $j = j + 1;$
    `EndWhile`
}

The proof of this result follows using the same argument as in the standard proof of correctness of the Ford-Fulkerson Algorithm. (See, for instance, [2].) Namely, one argues inductively that each $\mathcal{P}_j$ induces a minimum cost collection of arc-disjoint directed (s,t) paths. Thus the auxiliary digraph D_j contains no negative cycles (i.e., directed cycles with negative total cost). Clearly this holds for $j = 1$. Moreover, $\mathcal{P}_{j+1}$ is obtained by incrementing along the cheapest path Q_j in D_j, as in the Ford-Fulkerson Algorithm. One then deduces that D_{j+1} contains no negative cycles, and hence $\mathcal{P}_{j+1}$ is again a minimum cost collection of $j+1$ arc-disjoint paths.

For $j = 2, \ldots, \kappa(D)$, let z^j be the reservation vector assigning capacity $T/(j-1)$ to each arc in each of the paths in $\mathcal{P}_j$, and capacity 0 to each other arc. (This extends the notation of the previous section.) As discussed, an optimal solution to the problem of finding a minimum cost T-resilient diverse-paths reservation can always be found amongst the z^j's (we will see later that their costs are again unimodal in j).

We close this section by considering the problem of finding a minimum cost T-resilient integral reservation whose support is a collection of diverse (arc-disjoint) paths. Thus we denote by IDP the problem each instance of which consists of a network with two specified nodes s, t together with nonnegative integer costs on the arcs and an integer T. A (optimal) solution for an instance will be a minimum cost T-resilient diverse $s-t$ dipath capacity reservation which is integral.

The results of the previous section once again show that the support of an optimal such solution will consist of a collection of arc-disjoint $s-t$ dipaths $P_1, P_2, \ldots, P_j$ 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 $T-(j-2)r$.[4] We now show that the subproblem of IDP with the input T fixed to 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 [12] show that this problem is NP-complete.

Suppose that we are given such 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 will have cost zero. This is our instance of 3-IDP. From the preceding section we may deduce that an optimal 3-resilient reservation on diverse paths will either have support on (i) 2 arc-disjoint paths, in which case capacity 3 is reserved on each of the arcs of these paths, or (ii) 3 arc-disjoint 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 arc-disjoint 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 solution to 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 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 2-disjoint path problem has a solution.

Note that the above result shows that approximating 3-IDP to within a factor of $(1+\frac{1}{14})$ is NP-hard. On the other hand, Proposition 6 implies that applying the rounding procedure to an optimal fractional solution, will yield a $(1+\frac{1}{3})$-approximation to the optimal integral solution. We now improve this latter bound.

We have a polynomial time algorithm $\mathcal{A}$ that finds, for each value of j, *some* solutions $z^{j,-}$ and $z^{j,+}$, and chooses the best among these solutions. The algorithm 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

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

even minimum cost) set of l paths. All that we know, is that an optimal solution for an instance of IDP will have the *same form* as some $z^{l,-}$ or $z^{l,+}$ for some l, since it will arise 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 $O_F \leq O_I \leq O_A$. We prove the following result.

Theorem 9 *The algorithm $\mathcal{A}$ is a $\frac{1}{14}$-approximate algorithm for* IDP*, that is, for each instance: $O_A \leq \frac{15}{14} O_I$. Moreover, there is no ϵ-approximate algorithm for* IDP *with $\epsilon < \frac{1}{14}$.*

The latter statement, of course follows from the previous example showing that the decision version of IDP is NP-complete. It is worth noting 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 T. Many of these are ϵ-approximate algorithms with $\epsilon < \frac{1}{14}$. Indeed, Proposition 6 tells us that, for each T $O_A < (1 + 1/T)O_F \leq (1 + 1/T)O_I$. This is already a stronger result than Theorem 9 for $T \geq 14$.

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

Proof. Suppose an optimum integer solution z^* uses paths $P_1, \ldots, P_l$, each with reserved capacity r, and path P_{l+1}, with reserved capacity y. We may assume that $0 \leq y \leq r$, $l \geq 2$, and that r is either $\lfloor T/(l-1) \rfloor$ or $\lceil T/l \rceil$, with $T = (l-1)r + y$, since otherwise there is a solution of one of the given forms with no greater cost using the same set of paths, by Theorem 5. We also have that $\lfloor T/(l-1) \rfloor \geq \lceil T/l \rceil$.

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

The values of T and r determine y and l and, as we noted before the proof, we may assume that $T \leq 13$ and therefore $r \leq \lceil T/2 \rceil \leq 7$. 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 shall rule all 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 $\lfloor T/(l-1) \rfloor$ or $\lceil T/l \rceil$ for any integer l. In particular, we may assume that we have $T \geq 3$.

We require a lower bound on the cost of z^*. Notice that z^* can be written as y times the characteristic vector of some set of $l+1$ 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

$$O_I = C(z^*) \geq yC_{l+1} + (r-y)C_l.$$

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

$$C_{l+1} \geq \frac{l+1}{lr+y}O_A = \frac{l+1}{T+r}O_A.$$

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 $r' = \lceil T/(l-1) \rceil$; there is some integer solution with reserved capacity r' on the first $m < l$ paths from C_l, and $v \leq r'$ 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 good 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 6.

So we have $O_A \leq (T+r')C_l/l$, i.e.,

$$C_l \geq \frac{l}{T+r'}O_A.$$

We conclude that

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

After a little manipulation, this becomes

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

We could now run through all the 23 cases separately and show that $G \leq 1/15$ in each case. Let us proceed slightly more systematically.

First, we consider all cases with $l = 2$. 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 $G = 2y/(4y+2)(3y+2) \leq 1/15$ for $y \geq 1$.

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

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

3 Flow-Reservations and (T, M)-Resilience

In the previous section, we considered what happens when we impose the very strong condition that the reservation consist of a set of diverse paths. In this section, we ask what happens when we impose the somewhat weaker condition that our reservation vector be a flow vector – we call such a vector a *flow-reservation*. In other words, we are looking to find the minimum cost flow-reservation that satisfies the relevant resilience condition. In this section we assume that c is a nonnegative vector of arc costs.

We start this section by showing that the restriction to flow-reservations does not in general make the weak resilience problem any easier.

Suppose that s, t are source and sink respectively in D. Let D^e be the extension of D, obtained by adding the arcs (v, s) and (t, v) for each $v \in V - \{s, t\}$. Extend c to a vector c^e of costs on $A(D^e)$ by setting the cost of each new arc to 0. For any $x \in \mathcal{R}(T, D)$ let x^e be the extension of x to a flow vector in D^e (see Section 1.2). Then $c^e x^e = cx$. Conversely if x' is a flow vector in $\mathcal{R}(T, D^e)$, then restricting x' to the arcs of D yields a vector in $\mathcal{R}(T, D)$. Thus finding the minimum cost T-resilient flow reservation in D^e is equivalent to finding the minimum cost T-resilient reservation in D.

On the other hand, it turns out that weak, strong and net resilience are all equivalent for acyclic flow-reservations. Given that minimum cost flow vectors are acyclic, and that a flow-reservation is likely to constitute an initial working flow, this is not an improbable scenario.

Lemma 10 *Let x be an acyclic weakly T-resilient flow-reservation in a network D. Then x is net T-resilient.*

Proof. Suppose that x is an acyclic flow of value M, and consider an ordering $s = v_1, v_2, \ldots, v_n = t$

of V such that all arcs in the support of x go from a lower numbered node to a higher. We consider two cases.

Firstly, if all arcs a have $x_a \le M - T$, then we claim x is net T-resilient. Indeed, for every (s,t)-set S and arc $a \in \delta^+(S)$, the net flow out of S in $D - a$ is exactly $M - x_a \ge T$, so x satisfies the net partial-T cut constraint for (S, a).

Conversely, if there is some arc $a = (v_i, v_j)$ with $x_a > M - T$, then let $S = \{v_1, \ldots, v_i\}$. The total flow out of S in $D - a$ is $M - x_a < T$, so x violates the partial-T cut constraint for (S, a).

Thus, either x is net T-resilient, or it fails even to be weakly T-resilient. □

We have not resolved whether the concepts of net and strong resilience coincide for flow reservations. This would be the case, if the following conjecture held.

Conjecture 3.1 *Let D be a digraph with source s and sink t and let c be a nonnegative vector of arc costs. Then there exists an optimal strongly T-resilient flow reservation x which is acyclic.*

3.1 (T, M)-Resilience

Our approach to the task of finding optimal T-resilient flow-reservations is to start with the equally practical problem of (T, M)-resilience. Recall that a vector x is weakly/strongly/net (T, M)-resilient if it is weakly/strongly/net T-resilient and is also an (s, t) flow of value M. We now investigate the structure of an optimal reservation.

We call an arc a *critical* for a vector $x \in \mathcal{N}(T, D)$ if there is a *tight* net partial T-cut constraint for a pair (S, a), i.e., $x(\delta^+(S) - a) - x(\delta^-(S)) = T$; otherwise a is called *non-critical*. In particular if a is critical, then $x_a > 0$. We start with some simple lemmas.

Lemma 11 *Let x be an acyclic M-flow in a network D. Then x is T-resilient if and only if $x_a \le M - T$ for every arc a. Furthermore, if x is T-resilient, then the critical arcs are exactly those arcs a with $x_a = M - T$.*

Proof. This is immediate from the proof of Lemma 10. □

The next result applies to general net resilient reservations.

Lemma 12 *For $x \in \mathcal{N}(T, D)$, let D' be the digraph with node set V and edge set $\{a \in I(x) : a$ is non-critical $\}$, and let G' be the graph obtained by ignoring the directions on arcs of D'. If G' contains a cycle, then $x = \frac{1}{2}(x^1 + x^2)$, where x^1, x^2 are distinct vectors in $\mathcal{N}(T, D)$. Moreover, if x is an M-flow, so are x^1 and x^2.*

Proof. Suppose the contrary, and take a cycle C in G'. Impose a direction on C, so that C consists of a set C^+ of arcs $a \in D$, and the artificial "reverses" of another set C^- of arcs of D. Fix some positive ϵ such that $x_a \geq \epsilon$ for every $a \in C^+ \cup C^-$, and also $x(\delta^+(S) - a) - x(\delta^-(S)) \geq T + \epsilon$ for every $a \in C^+ \cup C^-$ and every (s,t)-set S.

Now consider the following two vectors x^1 and x^2. Both are equal to x except on arcs of $C^+ \cup C^-$. For x^1, we add ϵ to the capacity x_a of every arc in C^+, and subtract ϵ from that of every arc in C^-. In x^2, we do the opposite, adding ϵ to the capacities of arcs in C^- and subtracting ϵ from capacities of the arcs in C^+.

Clearly $x = \frac{1}{2}(x^1 + x^2)$ and, if x is an M-flow, then so are x^1 and x^2. In particular, note that x^1 and x^2 are non-negative vectors.

It remains to be shown that x^1 and x^2 are net T-resilient. If (say) x^1 is not, then it violates the net partial T-cut constraint for some pair (S, e). However, the total net flow out of S is the same for x^1 as it is for x, so the only case where the net flow out of S in $D - e$ is decreased is when $e \in C^+$, in which case it is decreased by just ϵ, so we still have $x(\delta^+(S) - e) - x(\delta^-(S)) \geq (T + \epsilon) - \epsilon = T$ even in this case. □

For net (T, M)-resilient flow reservations x, the previous lemmas combine to tell us that either x is a convex combination of two other net T-resilient flow reservations (in other words, x is not a vertex of $\mathcal{N}(T, D) \cap \mathcal{F}(M, D)$), or the set of arcs a with $0 < x_a < M - T$ forms a forest (no cycles in the underlying undirected graph). In fact, we can say more.

Lemma 13 *Let x be an acyclic vertex of the polyhedron $\mathcal{N}(T, D) \cap \mathcal{F}(M, D)$. Then the undirected graph G induced by the set N of arcs a with $0 < x_a < M - T$ is either empty or forms a path between s and t. Furthermore there is some value q such that each arc a in N is either an arc in the direction of the path with $x_a = q$ or an arc in the opposite direction with $x_a = M - T - q$.*

Proof. We have seen that the graph G is a forest. We claim that no node other than s and t can have degree 1 in G: this will clearly imply that G is a path.

Let v be any node other than s and t, having degree 1 in G. Then for every arc a incident with v other than one, the reserved capacity x_a is either 0 or $M - T$. For the remaining arc, the reserved capacity is strictly between 0 and $M-T$. Therefore the net flow out of v is not a multiple of $M-T$, so in particular is not 0. This is a contradiction.

The final observation follows by looking at flow conservation at nodes of the path, modulo $M-T$. □

We are now ready to show how to use the algorithm PATHS to find an optimal net (T,M)-resilient reservation. First let us note that, if $M < \frac{\kappa}{\kappa-1}T$, where $\kappa = \kappa(D)$ is the (s,t)-connectivity of the network, then no M-flow can be even weakly T-resilient. Indeed, for any (s,t)-cut of size κ, some arc in the cut must carry at least M/κ of the capacity, and if this arc fails then the remaining arcs have total capacity at most $(\kappa-1)M/\kappa < T$.

As before, for $j = 2,\ldots,\kappa$, let z^j be the reservation vector, generated via algorithm PATHS, obtained by taking the minimum cost collection of j (s,t) paths provided by the algorithm, and giving each arc in each of the paths capacity $T/(j-1)$. Use of the algorithm guarantees that each pair z^j, z^{j+1} differs only on the single augmenting path Q_j in the auxiliary digraph D_j. The algorithm also provides a minimum cost (s,t) path $\mathcal{P}_1$; let y^1 be the reservation vector obtained by giving each arc of $\mathcal{P}_1$ capacity 1.

The reservation vector z^j is net T-resilient for $j \geq 2$, and is a $\frac{j}{j-1}T$-flow. If M lies between $\frac{j+1}{j}T$ and $\frac{j}{j-1}T$, say $M = \lambda\frac{j+1}{j}T + (1-\lambda)\frac{j}{j-1}T$, with $0 \leq \lambda \leq 1$, then the vector $z^{(M)} = \lambda z^{j+1} + (1-\lambda)z^j$ is net (T,M)-resilient. For $M \geq 2T$, the vector $z^{(M)} = z^2 + (M-2T)y^1$ is net (T,M)-resilient.

Let us look at the structure of $z^{(M)}$ a little further. Suppose first that $\frac{j+1}{j}T \leq M \leq \frac{j}{j-1}T$, for some $j \geq 2$. Except on the augmenting path Q_j, all arcs in the support of $z^{(M)}$ have the same capacity, and this is $\lambda T/j + (1-\lambda)T/(j-1) = M-T$. From this, and the fact that $z^{(M)}$ is a flow of value M, we can deduce that forward arcs on the augmenting path Q_j have capacity $M - j(M-T)$, and reverse arcs have capacity $(j+1)(M-T) - M$. Similar comments apply for $M \geq 2T$.

Our claim is that the vectors $z^{(M)}$ are minimum cost net (T,M)-resilient reservations. Of course, a consequence is that the algorithm PATHS can then be used to generate an optimal solution efficiently.

Theorem 14 *Let D be a network with (s,t)-connectivity κ, equipped with non-negative costs on its arcs, and let M and T be such that $M \geq \frac{\kappa}{\kappa-1}T$. Then $z^{(M)}$ is a minimum cost net (T,M)-resilient reservation. If D is acyclic, then $z^{(M)}$ is also a minimum cost weakly/strongly (T,M)-resilient reservation.*

Proof. Let x^* be a minimum cost net (T,M)-resilient reservation (or, if D is acyclic, a minimum cost weakly/strongly (T,M)-resilient reservation), that is not a convex combination of two other net (weakly/strongly) (T,M)-resilient reservations. We may assume that x^* is acyclic (a fortiori, if D is acyclic).

Let N be the set of non-critical arcs. If $N = \emptyset$, then by flow-conservation and acyclicity, x^* may be written as $(M-T)\sum_{i=1}^{l}\chi^{P_i}$ where $P_1, P_2, \ldots, P_l$ are arc-disjoint directed (s,t) paths. This implies that $M = l(M-T)$. Also, since the vector $z^l = z^{(M)}$ arises in the same way from a minimum cost collection of l arc-disjoint paths, the cost of z^l is at most that of x^*, as claimed. Thus we will assume henceforth that N is nonempty.

Now we apply Lemma 13 to x^* and decompose the set of arcs of $I(x^*)$ into a set Q of arcs with capacity $M-T$ under x^*, a set F of arcs with some fixed capacity q, and a set R of arcs with capacity $M-T-q$, such that the set F, together with the reverses of the arcs in R, forms a directed path P from s to t.

Note that no arc into s is assigned positive value, since x^* is acyclic. In particular, the first arc a of P must be in F and so $q = x^*_a = M - l(M-T)$, where l is the number of critical arcs incident to s.

The flow x^* can be "augmented" along the path P by an amount $M-T-q$, to obtain an acyclic flow of value $(l+1)(M-T)$ with support $Q \cup F$, in which every arc of $Q \cup F$ has capacity $M-T$. Thus $Q \cup F$ can be decomposed as a set of $l+1$ arc-disjoint paths. Similarly we can decrement x^* along P by an amount q to obtain an acyclic flow of value $l(M-T)$ with support $Q \cup R$, and hence $Q \cup R$ can be decomposed as a set of l arc-disjoint paths.

Now, replacing $Q \cup F$ by the arcs in a minimum cost set $\mathcal{P}_{l+1}$ of $l+1$ paths, and $Q \cup R$ by the arcs in $\mathcal{P}_l$, leads us to another (T, M) resilient reservation of no higher cost. The remarks prior to the theorem tell us that this alternative solution is just $z^{(M)}$, as desired. $\square$

This gives the structure of a minimum cost net T-resilient M-flow. But what should we do if we just want the minimum T-resilient flow, i.e., if we are free to choose the value of M?

Since $z^{(M)}$ is a convex combination of some pair of the strongly T-resilient flows z^j, for $2T \geq M \geq \frac{\kappa}{\kappa-1}T$, and $z^{(M)}$ dominates z^2 for $M > 2T$, the minimum of the cost over all the $z^{(M)}$ is attained at some z^j. This gives us the following result.

Corollary 15 *If c is a vector of non-negative costs, then an optimal solution to*

$$\min\{cx : x \in \mathcal{N}(T, D),\ x \textit{ is a flow vector}\}$$

is given by $x = z^j$ for some $j \geq 2$, i.e., it is obtained by giving capacity $T/(j-1)$ to each arc in a minimum cost collection $P_1, P_2, \ldots, P_j$ of arc-disjoint directed (s,t) paths.

In other words, the optimal net T-resilient flow vector is a diverse-paths solution. In an acyclic digraph, the optimal weakly/strongly T-resilient flow vector is also a diverse-paths solution.

The extension to the k-failure case follows in a similar manner to that in Section 2. We omit the details.

3.2 Integrality

We now turn to the problem of finding minimum cost (net) T-resilient flow-reservations, where T is an integer and the allocations must be integers. Note that if such a flow is of value M, then M is necessarily an integer. Observe that, when M is an integer, the optimal (T, M)-resilient vectors $z^{(M)}$ are integral, so one of these will be optimal for the integral problem. There might yet be an exponential number of possible values of M to consider, but for the fact that, for each j, only the first and last $z^{(M)}$ on the segment between $z^{(j)}$ and $z^{(j+1)}$ can be optimal. Thus it is enough to find the minimum cost T-resilient M-flow for each M of the form $\lfloor \frac{jT}{j-1} \rfloor$ or $\lceil \frac{jT}{j-1} \rceil$ for each $j = 2, \ldots, \kappa$, where κ is the (s,t)-connectivity. Using algorithm PATHS, this can be achieved in $O(\kappa mn)$ time, where m, n denote the number of arcs and nodes of D respectively. Furthermore, as in the case when the paths are pre-specified, we now show that the cost of $z^{(M)}$ is unimodal in M, so that we may stop the process as soon as the costs start to go up. Alternatively, we can find the optimal solution z^{j_0} among the z^j, and then we know that the optimal integer solution will be $z^{(M)}$, where M is either $\lfloor \frac{j_0 T}{j_0-1} \rfloor$ or $\lceil \frac{j_0 T}{j_0-1} \rceil$.

For $j = 2, \ldots, \kappa(D)$ let z^j be the reservation assigning capacity $T/(j-1)$ to each arc in each of the paths in the family P_j (as generated via algorithms *Paths* in Section 2.3) and capacity 0 to each other arc.

Proposition 16 *Let j, h and k such that $2 \leq j < h < k \leq \kappa(D)$. If $cz^j < cz^h$ then $cz^h \leq cz^k$.*

Proof. Suppose the contrary: there exists j, h and k, $j < h < k$ such that $cz^j < cz^h$ and $cz^h > cz^k$. Let $M = T\frac{h}{h-1}$ and let $0 < \lambda < 1$ such that $\frac{h}{h-1} = \lambda\frac{j}{j-1} + (1-\lambda)\frac{k}{k-1}$. Then $z' = \lambda z^j + (1-\lambda)z^k$ is a (T, M) resilient flow, and of course $cz' \leq \max\{cz^j, cz^k\}$. But from Theorem 14 we know that $cz^h \leq cz'$, which is a contradiction. □

We close by giving a small example - Figure 4 - which shows an optimal integral flow-reservation which is not a diverse path solution, and, for which the optimal diverse path solution does not correspond to a minimum cost collection of j paths for any j. One checks that the optimal fractional j diverse path solutions are $(j = 2)$ assign T to the two cheapest arcs between s and t, for a cost of $10T$, $(j = 3)$ assign $T/2$ to the two cheapest arcs again, as well as on the path $s - u - v - t$ for a cost of $8T$, $(j = 4)$ uses the paths $s - u - t$ and $s - v - t$ and costs $8T$ and $(j = 5)$ the optimal solution has a cost of $33T/4$.

Now suppose that $T = 2P + 1$, for some $P \geq 6$. Then an optimal integer flow-reservation is obtained by reserving capacity P on the two cheapest (s,t) arcs as well as on (s,u) and (v,t), then $P - 1$ on the arc (u,v) and 1 on each of (s,v) and (u,t). This costs $16P + 8$ and is obtained by augmenting from the $(j = 3)$ towards the $(j = 4)$ solution (which was obvious since both of z^3, z^4 were optimal for the fractional problem).

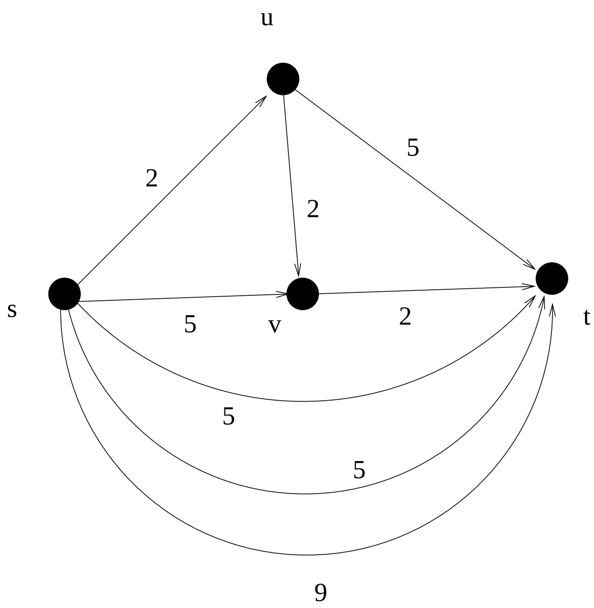


Figure 4:

On the other hand, the best integer diverse path reservation using the minimum cost collection of 3 paths costs $16P + 10$. And the mincost collection of 4 paths leads to an integer diverse path solution of $17P + 7$. But assigning 1 to the expensive (s, t) arc, and assigning P to the other two, as well as to the arcs of the path $s - u - v - t$ yields a solution of cost $16P + 9$.

4 Resilient Reservations with no Restrictions

In this section we consider the general problem of finding minimum cost weak/strong/net resilient vectors where there are no added constraints imposed on the reservation vectors themselves. Our results in this section are mostly negative, but we hope they give a flavour of the difficulties that have to be overcome in tackling the general resilience problems.

We start with a result showing how beastly the notion of strong resilience becomes when we leave the domain of flow-reservations. In fact, we now show that the problem of even recognizing whether a vector x is strongly resilient is coNP-complete (even if $I(x)$ induces an acyclic bipartite graph). To see this, we show that the following problem is NP-complete.

Problem SUBFLOW. Let $D = (V, A)$ be a digraph, s and t two vertices of D, T an integer, and $x \in \mathbb{Q}^A$. Does there exist an arc a, and a maximal (s, t) flow vector $y \leq x$ in $D - a$ whose value is less than T.

In the following, if D is a digraph for which a capacity function x is defined, let us denote by $f(D)$ the minimum value of a maximal (s,t) flow vector.

Clearly, SUBFLOW is in NP, so we will reduce the problem of finding a minimum maximal matching (MMM) to SUBFLOW. MMM is known to be NP-hard even on bipartite graphs [20]. Let $G = (V_1 \cup V_2, E)$ be a bipartite graph and let us denote by $\beta(G)$ the cardinality of a minimum maximal matching for G. We associate with G the following capacitated network D: $V = V_1 \cup V_2 \cup \{s,t\}$, $A = \{(u,v) : uv \in E, u \in V_1, v \in V_2\} \cup \{(s,u) : u \in V_1\} \cup \{(v,t) : u \in V_2\} \cup \{st\}$. Let x be the vector of all 1's in $\mathbb{Q}^A$ and consider the instance I of SUBFLOW consisting of D, s, t, x and some integer T.

First of all, note that I is a no-instance of SUBFLOW if and only if for each arc a, $f(D-a) \geq T$. One sees trivially that $f(D) = f(D-st)+1$; on the other hand for each $a \in A$ we have that $f(D) \leq f(D-a)+1$. Hence $f(D-st) \leq f(D-a)$ for each $a \in A$. Thus we have a no-instance of problem SUBFLOW if and only $f(D-st) \geq T$.

Now, clearly, it is enough to show that $f(D-st) = \beta(G)$. First of all, trivially $f(D-st) \leq \beta(G)$. Now let us show that $f(D-st) \geq \beta(G)$. Let $y \leq 1$ be an (s,t) flow vector of $D - st$ such that the value of y is $f(D-st)$ and there exists no feasible flow vector $z \leq 1$ which properly dominates y. We claim that there exists such a vector y which is integral. In this case it induces a maximal matching of G with the same value and so $f(D-st) \geq \beta(G)$.

To see this, note that if we remove the set $\mathcal{T}$ of arcs a with $y_a = 1$, then there is no directed path from s to t, otherwise we could augment y by a positive amount. Now solve the maximum flow problem in $D - st$ with upper bounds of 1 on each arc, and lower bounds of 1 on the arcs of $\mathcal{T}$. The vector y is feasible for this flow problem and hence by standard flow results, there exists an integral such flow vector. This completes the proof. □

In contrast, one may optimize over $\mathcal{N}(T,D)$ by solving a linear program with a polynomially (in the size of D) bounded number of constraints and variables. We describe this now. We have seen (Lemma 2) that without loss of generality s,t are a source and sink respectively in D. Consider the extended digraph D^e obtained by adding the arcs $l_v = (v,s), r_v = (t,v)$ for each node $v \neq s,t$. Let $x \in \mathcal{N}(T,D)$ and let x^e be the extension of x in D^e. Set $M = val(x^e)$. We must have:

$$x^e_{(u,v)} \leq (M-T) + x^e_{r_u} + x^e_{l_v}. \tag{8}$$

otherwise one sees that the net partial T-cut constraint for $(\{s,u\},(u,v))$ is violated. Conversely, suppose that x^e is an M-flow in D^e which satisfies the *sidebar* constraints (8) and let x be the vector restricted to $\mathbb{Q}^A$. Suppose in D there exists an (s,t)-set S and an arc $(u,v) \in \delta^+(S)$ such that $x(\delta^+(S) - uv) - x(\delta^-(S)) < T$. Now consider the same cut in D^e (we dispense with subscripts). We know that $x^e(\delta^+(S)) - x^e(\delta^-(S)) = M$. On the other hand, $x(\delta^+(S)) - x(\delta^-(S)) = x^e(\delta^+(S)) - x^e(\delta^-(S)) + \sum_{z \in S} x^e_{r_z} + \sum_{z \in V-S} x^e_{l_z} = M + \sum_{z \in S} x^e_{r_z} + \sum_{z \in V-S} x^e_{l_z} \geq M + x^e_{r_u} + x^e_{l_v}$. But then

$x(\delta^+(S)) - x(\delta^-(S)) \geq\ M + x_{r_u} + x_{l_v}$ and by the sidebar constraint, this is at least $T + x_{uv}$.

We have thus defined a min cost flow problem plus some extra sidebar constraints and an extra variable M for the flow value. Thus $\mathcal{N}(T, D)$ is the appropriate projection for the set of all such flows in D^e:

$$\mathcal{N}(T, D) = \{x : \exists x', M \ \ s.t.(x, x') \text{ is an } M\text{-flow in } D^e \text{ which satisfies (8) }\}.$$

4.1 Examples of Vertices of the Polyhedra $\mathcal{R}(T, D)$ and $\mathcal{N}(T, D)$

We present several examples of vertices for the polyhedra $\mathcal{N}(T, D)$ and $\mathcal{R}(T, D)$. One advantage for flow-reservations was that the optimal solutions were easy to visualize, since they corresponded to collections of diverse paths. In the general case we will now see this is not the case.

By definition, x is an extreme point of $\mathcal{N}(T, D)$ if (i) $x \in \mathcal{N}(T, D)$ and (ii) there is some subset of $|A|$ linearly independent constraints, from the system of non-negativity and net partial T-cut constraints, which are satisfied with equality by x. Recall that an arc a is termed critical if there is some tight net partial T-cut inequality for x which corresponds to a pair (S, a). Similar remarks hold for extreme points of the polyhedron $\mathcal{R}(T, D)$.

Let us begin by giving an example which shows that vertices may not correspond to assigning appropriate weights to a collection of diverse (arc-disjoint) paths. The weights assigned to the arcs in D given in Figure 5 correspond to a vertex both of $\mathcal{N}(7, D)$ and of $\mathcal{R}(7, D)$. To see this, one considers seven partial cuts associated with $\delta^+(s)$ as well as four more cuts associated with the two directed cuts which contain exactly two arcs of weight 3. A different example for which the graph does not decompose as a 'two-sum' is given in Figure 6. The vector here is a vertex both of $\mathcal{N}(4, D)$ and of $\mathcal{R}(4, D)$.

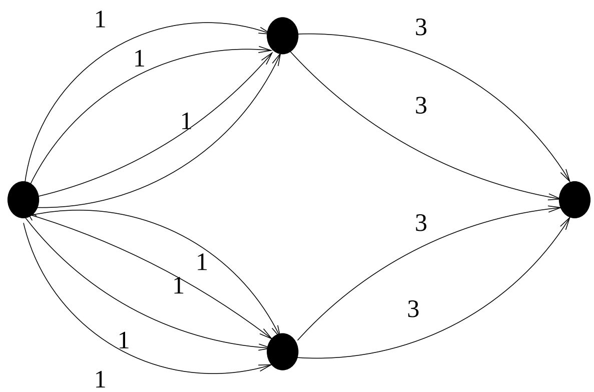


Figure 5: *A vertex of* $\mathcal{R}(7, D), \mathcal{N}(7, D)$

Indeed, for weak and net resilience, basic optimal solutions may have components of the form $\frac{pT}{q}$ for

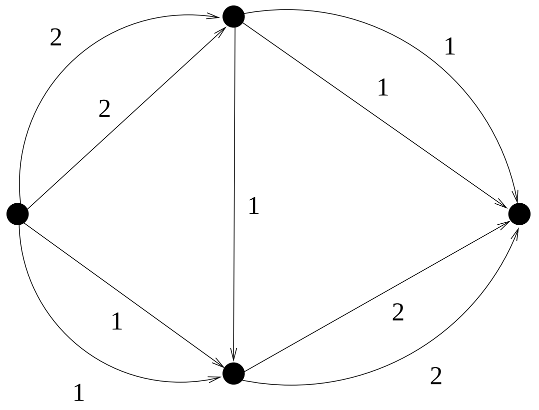


Figure 6: *A vertex of* $\mathcal{R}(4, D), \mathcal{N}(4, D)$

any $0 \leq p/q \leq 1$. Indeed for any such p, q, Figure 7 displays such an extreme point. In this example, for appropriate costs (e.g., all arcs having the same cost except for two extremely cheap arcs from s to the intermediate node), an optimal solution will assign $\frac{T}{q}$ to each arc into t and the remaining pair of arcs will be assigned the value $\frac{pT}{q}$. Note that the cut induced by $V - t$ will have all of its arcs critical and that the two arcs with value $\frac{pT}{q}$ are critical for $\delta^+(s)$.

In the case of net resilience we do not know of any vertices x^* with non-critical edges in the support. We conjecture the following.

Conjecture 4.1

- *If x^* is a vertex of $\mathcal{N}(T, D)$, then each arc $a \in I(x^*)$ is critical for x^*.*
- *Let $c \geq 0$ and x^* be a unique optimal solution for* $\min\{cx : x \in \mathcal{N}(T, D)\}$. *If $\phi(x^*) = M$ and $x^* \neq x^e$, then there is no (s, t) flow x of value M for which $x_a \leq M - T$ for each arc a.*

By contrast, our final example – Figure 8 – shows an extreme point x^* of $\mathcal{R}(T, D)$ which is not in $\mathcal{N}(T, D)$ and for which there are non-critical arcs a (the two arcs not incident with s and t) with $x^*_a > 0$. This example is also a basic solution for the node-deletion version of the problem; this shows that optimal node-T-resilient reservations are not always diverse-paths reservations.

One hope of understanding the structure of the extreme points x of $\mathcal{R}(T, D)$ is to understand the role of the critical arcs, that is, the arcs a for which there exists an (s, t)-set S such that $x(\delta^+(S) - a) = T$. We will call such a set S *tight* if this holds for one of the arcs in $\delta^+(S)$. We have seen that it need not be the case that all arcs in the support are critical for an extreme point of $\mathcal{R}(T, D)$. On the other hand, if x is extreme, then there must be a large number of such tight constraints defining x. One can show that if a is a critical arc, then there is no tight set S for which $a \in \delta^-(S)$. Suppose this were the case and let e be a critical arc for S and let S' be a second (s, t)-set for which

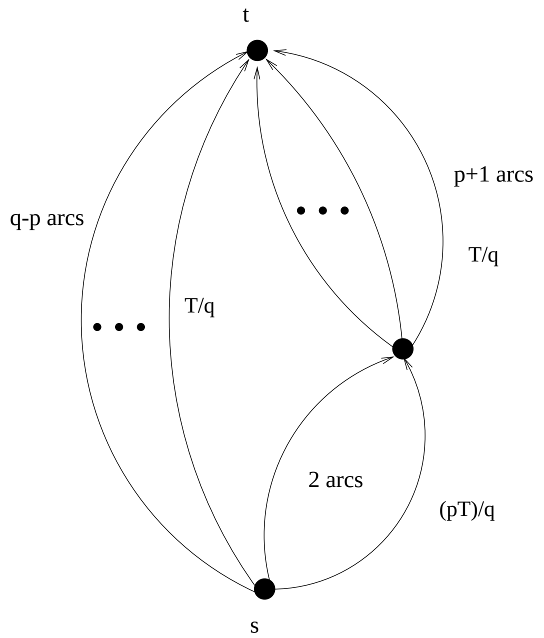


Figure 7: *A vertex of* $\mathcal{R}(T,D), \mathcal{N}(T,D)$

$x(\delta^+(S') - a) = T$. Then $x(\delta^+(S)) + x(\delta^+(S')) = 2T + x_e + x_a$. But the left hand side is also equal to $x(\delta^+(S \cap S')) + x(\delta^+(S \cup S')) + x[S', S] + x[S', S]$. In particular, $a \in [S', S]$. Consider the case where $e \in \delta^+(S \cup S)$. Then since $x(\delta^+(S \cap S')) > T$, we have that $x(\delta^+(S)) + x(\delta^+(S')) > T + (T + x_e) + x_a$, a contradiction. The other cases follow similarly. One consequence of this, is that if all arcs in $I(x)$ are critical, then $I(x)$ is acyclic. Unfortunately, further understanding of the structure of these extreme points has eluded us. We leave two conjectures.

Conjecture 4.2 *Let x be an extreme point of $\mathcal{R}(T,D)$.*

- *There exists a directed path from s to t consisting only of critical arcs.*
- *There exists an (s,t)-set S such that the reserved capacities on all arcs in $I(x) \cap \delta^+(S)$ have the same value, and each arc in this set is critical.*

We believe that these may contribute to the resolution of the main open problem we leave: to find an algorithm for weak resilience which uses some generalization of the pivot operation (i.e., cycle augmentations) in standard minimum cost flow. More generally, we ask for a combinatorial algorithm for this problem.

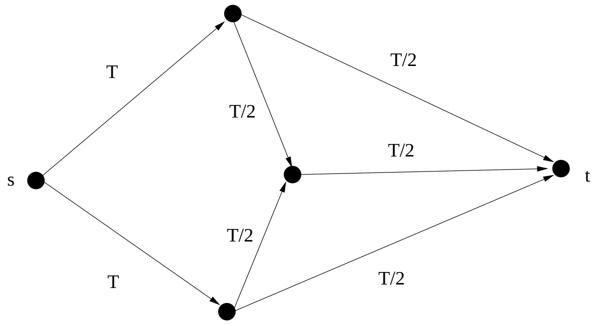


Figure 8: *A basic solution for the node-deletion version*

4.2 NP-completeness of Integer Resilience Problems, and an Approximation Algorithm

We next show that the problems of finding a minimum cost integral T-resilient – weak, strong or net – vector are all NP-hard. Note that the remarks at the beginning of Section 3 also imply that the problem of finding a mincost integral weakly resilient *flow-reservation* is NP-hard (in contrast to the integral net or strongly resilient flow-reservation problems, for which we gave a polynomial time algorithm).

Here is the integer weak resilience problem, couched as a decision problem.

INTEGER WEAK RESILIENCE

Instance: a digraph D, with integer costs c_{ij} on the arcs, with a single source s and destination t, a target resilience 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 weakly T-resilient?

Theorem 17 INTEGER WEAK RESILIENCE *is strongly NP-complete.*

Proof. Certainly the problem is in NP, since checking T-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 WEAK RESILIENCE with $m + 3n + 2$ nodes, $11m + 3n$ arcs each of cost 1, n, or $2n$, such that there is a weakly $(4m + 3n - 1)$-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.

Consider any weakly T-resilient integer reservation vector x of cost at most $(2n+1)(T+1)+n$, where $T = 4m+3n-1$. First note that the reservation x must dominate a flow of value $T+1$ since otherwise there is a cut of capacity at most T. Look first at the set A_0 of "expensive" arcs not incident with s. A flow of value $T+1$ necessarily costs $2n(T+1)$ on A_0. Thus the cost of x on A_0 is at least $2n(T+1)$, and equality is only possible if every arc in A_0 has reserved capacity just 1. Thus on A_0, the reservation looks like the characteristic vector of a set of $T + 1 = 4m + 3n$ arc-disjoint paths. Such a set of paths must include every arc arriving at t, and uses 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. The cost of any integer T-resilient reservation not of this form must be at least $2n(T + 1) + n$ on these arcs.

Now consider the reservation vector x on the set A_1 of arcs leaving s. Certainly the total cost of the reservation on A_1 is greater than $T+1$, since some arc must have reserved capacity at least 2. Thus we see that, since the total cost of x is at most $(2n+1)(T+1)+n$, then the reservation on A_0 does indeed form a set of $T + 1$ arc-disjoint paths, with $d(v) \in \{4, 5, 6, 7\}$ starting from each $v \in \mathcal{T}$. 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 $d(v) = 4$, since then only 1 need be reserved on each arc. For $d(v) > 4$, one routinely checks that such a reservation costs at least $d(v) + 1$, since arcs of capacity at least 2 are needed. So the total cost of x on A_1 consistent with the values $d(v)$ is at least $\sum_v d(v)$ plus the number N of elements v for which $d(v) > 4$. We know that $\sum_v d(v) = T + 1 = 4m + 3n$, $|\mathcal{T}| = m$, and $d(v) \leq 7$ for all v. Therefore $N \geq n$, with equality if and only if $d(v) = 7$ for just n elements of $\mathcal{T}$, and $d(v) = 4$ for the remainder. In this case, the cost of x on A_1 is $T + 1 + n$, and otherwise is at least $T + 2 + n$.

Hence, if there is such a reservation vector x of total cost just $(2n + 1)(T + 1) + n$, then there are just n elements $v \in \mathcal{T}$ with $d(v) = 7$, while the remainder have $d(v) = 4$. A collection of $T + 1$ arc-disjoint paths on A_0 subject to these conditions must involve n elements of $\mathcal{T}$ for which the $3n$ arcs from these elements to $A \cup B \cup C$ go to distinct nodes, i.e., a 3D-matching in the original instance.

Conversely, given a 3D-matching $\mathcal{U}$, we can find a weakly T-resilient reservation by reversing the argument. We reserve 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 weakly T-resilient, and has the required cost. $\square$

Of course, the requirement that all reservations be integers is crucial. However, notice that the reservation we emerge with is actually both strongly and net resilient as well, so the corresponding problems for strong and net resilience are also NP-hard.

On the positive side, an extremely simple 1-approximate algorithm for weak resilience (in the fractional or integral versions) is to find a minimum cost pair of edge-disjoint paths P_1, P_2, and to then output the reservation vector $z^2 = T(\chi^{A(P_1)} + \chi^{A(P_2)})$. To see this, it will be enough to show the following.

Proposition 18 *If $x \in \mathcal{R}(T, D)$ and $c \geq 0$ is a vector of arc costs, then $c \cdot x \geq \frac{T}{2}(\chi^{A(P_1)} + \chi^{A(P_2)})$.*

In order to prove this we require a result of Aggarwal and Orlin [1]. Let $\mathcal{O}(j, D)$ denote the dominant of the convex hull of the vectors of the form $\sum_{i=1}^{j} \chi^{A(P_i)}$ where $P_1, \ldots, P_j$ form a collection of arc-disjoint directed (s, t) paths.

Theorem 19 (Aggarwal,Orlin) *A flow vector x of value j is in $\mathcal{O}(j, D)$ if and only if $x_a \leq 1$ for all $a \in A(D)$.*

We note that this result is very closely related to, and can be derived from, a theorem of Egawa, Kaneko and Matsumoto [10].

Proof of Proposition 18. Let x' be obtained from x by setting $x'_a = \min\{T/2, x_a\}$. We claim that $x'(\delta^+(S)) \geq T$ for all (s, t)-sets S. If this is not the case, then there is some violated S for which $\delta^+(S)$ contains an arc a such that $x_a > T/2$. Now since $x(\delta^+(S) - a) \geq T$, there must be at least one other arc $b \in (\delta^+(S) - a)$ which was also truncated. Thus $x'_b = T/2$ and so $x'_a + x'_b = T$, a contradiction.

Now let f be any (s, t) flow of value T which satisfies $f \leq x'$. Then $c \cdot f \leq c \cdot x$. Moreover, since $f_a \leq T/2$ for each arc, $f = \frac{T}{2} g$ where $g \in \mathcal{O}(2, D)$, and so $g \geq \chi^{A(Q_1)} + \chi^{(Q_2)}$ for some pair (Q_1, Q_2) of arc-disjoint directed (s, t) paths. Thus $c \cdot g \geq c \cdot (\chi^{A(Q_1)} + \chi^{(Q_2)}) \geq c \cdot (\chi^{A(P_1)} + \chi^{A(P_2)})$, and the proof is complete. $\square$

The constant 2 in Proposition 18 is best possible. Indeed, consider the following example. The network D has three nodes s, u and t. There are two arcs of cost 1 from s to u, and r arcs of large cost c from u to t. The minimum cost weakly 1-resilient reservation involves giving capacity 1 to the arcs between s

and u, and capacity $\frac{1}{r-1}$ to the arcs between u and t, at a cost of $2+cr/(r-1)$. However, the minimum cost diverse-paths reservation, z^2, uses only two of the arcs from u to t, at a total cost of $2+2c$. For large c and r, the ratio between these two costs can be made arbitrarily close to 2: the optimal diverse-paths reservation is almost twice as expensive as the optimal weakly 1-resilient reservation.

As usual, one would expect that the normal situation is even better than suggested by Proposition 18, and in many practical settings the optimal diverse-paths reservation (which can be found easily using the methods of Section 3) will not be very far from the overall optimal solution in cost. Given the simplicity of a diverse-paths solution, it is likely that this is a good solution to adopt.

5 Applications to More than one Source-Destination Pair

Consider the problem where we are required to reserve capacity in a network (directed graph) for traffic between all pairs of nodes. Our objective is to minimize the total cost of the reservations we make, subject to supporting a given target traffic matrix $[M_{ij}]$ (i.e., for each pair i, j, there is enough capacity reserved for this pair to dominate a network flow of value M_{ij} from i to j). If we let $\mathcal{P}_{ij}$ denote the set of paths connecting node i to node j, then we may view this as a problem, say **Q**, of assigning nonnegative weights w_P to the directed paths P of the network, and integer weights y_a to the arcs of the network such that: (i) $\sum_{P\in\mathcal{P}_{ij}} w_P = M_{ij}$: the sum of weights assigned to paths connecting node i to node j is at least the demand M_{ij} and (ii) $\sum_{a\in A(P)} w_P \leq y_a$: the capacity y_a assigned to arc a is enough to carry the traffic assigned to paths through a. Subject to (i),(ii) we wish to minimize the capacity cost $\sum_a c_a y_a$.

Evidently, **Q** may be solved trivially by finding a shortest path between each pair of nodes, and reserving capacity M_{ij} on the arcs of this path. The key point here is that there is no interaction between traffic from different demand pairs, and so each pair can be treated separately. In this paper, we have discussed solution techniques for solving a resilient version(s) of **Q**, where each reservation vector must satisfy a specified level of resilience.

The problem **Q** itself is an oversimplified prototype for a network design problem. This problem becomes computationally more difficult (in practice) as soon as we move to the a scenario where capacity is reserved in bundles of traffic units, that is, where the constraints (ii) become $\sum_{a\in A(P)} w_P \leq y_a U_a$ for integers U_a on the arcs. In this case, there may be considerable interaction between traffic from different demand pairs and so the overall problem may no longer be solved one pair at a time. In this section, we discuss this version of the problem where in addition the capacity reservations must be resilient. We note that a close relative of **Q** is the *multicommodity flow problem* - the feasibility problem obtained by fixing the y_a's, and the w_P's are nonnegative real numbers. This problem is polynomially solvable by general linear programming algorithms. The version where the w_P's are required to be integers is

NP-hard (2DIV-PATHS is a special case).

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 failure rates T_i, $i = 1, 2, \ldots, q$. Each *commodity* (s_i, t_i) must reserve capacity x^i in a network D which is T_i-resilient for the source-destination pair (s_i, t_i).

One approach to tackling this problem is to insist that the reservation vectors are strongly resilient flow-reservations. In the absence of other commodities, we have seen in Section 3 that such reservations correspond to diverse-path reservations. We have seen that such reservations may cost more, even in the 1-commodity case, but they do have several advantages. For a start, the diverse-paths routing may be "hardwired" at the terminal end nodes s_i, t_i, so that traffic routing complexity is diminished. If the network manager controls traffic flow, then this allows the even splitting 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. The final advantage to this approach is that we may formulate the problem in a similar manner to the well-known multicommodity capacity allocation problem where there is a flow demand between each commodity pair and one must find routings of the flow such that capacity cost in the network is minimized.

To see this, suppose first that for a commodity i, we restrict attention to $\frac{j_i T_i}{j_i - 1}$-flows in building our resilient reservation vector. From Section 3.1, we then know that we must search for a (s_i, t_i) flow vector r^i of value j_i and then place capacity $r^i_a \frac{T_i}{j_i - 1}$ on each arc a. One may solve this as a mixed integer program as follows.

$$
\begin{array}{ll}
\min c_a U_a & \\
\sum_{i=1}^{q} \frac{T_i}{j_i - 1} r^i_a \leq x_a U_a & \text{for each arc } a \\
r^i(\delta^+(v)) = r^i(\delta^-(v)) & \text{for each } i \text{ and } v \neq s_i, t_i \\
r^i(\delta^+(s_i)) - r^i(\delta^-(s_i)) = j_i & \text{for each } i \\
0 \leq r^i_a \leq 1 & \text{for each } a \text{ and } i \\
x_a \geq 0 & \text{an integer for each arc } a
\end{array}
$$

Note that in this formulation we allow the reservation flow vectors r^i to take on fractional values, but that the number x_a of copies of arc a, as well as the capacity U_a on each copy, are forced to be integers. Note that this formulation will yield flow vectors r^i which can be decomposed into a linear combination of vectors of the form $\sum_{l=1}^{j_i} \chi^{A(P^i_l)}$ where the P_l's form an arc-disjoint collection of directed (s_i, t_i) paths (by Theorem 19). This detracts from the simplicity outlined as an advantage of using diverse-paths reservations, if this decomposition requires more than one such collection of diverse paths. This can be alleviated by forcing the flow variables r^i to be $0 - 1$ valued and then solving the pure integer program. Another drawback is that, for each commodity, there are several choices of j_i, and we must try each of these separately within a branching routine (solving a network capacity allocation problem each time!). As in the problem **Q**, if the U_a's are all 1 and the reservations are required to be integers, then we may solve this problem one pair at a time using the methods of Section 2.

References

[1] C.C. Aggarwal, J.B. Orlin, On multi-route flows in networks, *Sloan School Working Paper*, 3954-97 (1997).

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

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

[4] D. Alevras, M. Grötschel, R. Wessäly, Cost-efficient network synthesis from leased lines, *Konrad-Zuse-Zentrum für Informationstechnik Berlin, Technical Report* No. 97-22 (1997).

[5] D. Bienstock, G. Muratore, Strong inequalities for capacitated survivable network design problems, *Columbia IEOR Tech Report*, Dec. 1997.

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

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

[8] 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.

[9] C. Colbourn, *The Combinatorics of Network Reliability*, Oxford University Press, 1987.

[10] Y. Egawa, A. Kaneko, M. Matsumoto, A mixed version of Menger's Theorem, *Combinatorica* **11** (1991) 71-74.

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

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

[13] 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.

[14] M. Grötschel, L. Lovász and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization,* (2nd Edition), Springer-Verlag, Berlin, 1988.

[15] 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.

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

[17] A. Schrijver, *A Course in Combinatorial Optimization*, CWI and University of Amsterdam, 1997.

[18] P.D. Seymour, Sums of circuits, in J.A. Bondy, U.S.R. Murty, *Graph Theory and Related Topics*, Academic Press, New York, 1978 341-355.

[19] 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.

[20] M. Yannakakis, F. Gavril, Edge dominating sets in graphs, *SIAM J. Appl. Math.* **48** (1980) 364–373.