Hitting-Set Algorithms for Fast Data Recovery in the Face of Geographic Correlated Attacks

Alon Efrat Esther Ezra Rom Pinchasi Swaminathan Sankararaman

***Abstract*—In distributed storage networks, ensuring data availability in the presence of hardware faults is an important requirement. Typically, redundancy schemes such as replication and erasure coding are used to ensure this. In case of hardware failures, these networks may be disconnected into multiple components each of which may require access to the data. In addition, the placement of redundant information must also be optimized as it is ever-changing and requires constant updating.**

We study the problem of selecting a set of nodes in networks of this kind so that data availability is maintained in the face of *geographically correlated failures*. We model failure events of arbitrary shapes as the union of disks or lines in the plane and present approximation algorithms for the problem of selecting a minimum number of redundant information locations (such as replicas or coded file segments) so that data recovery is guaranteed at every node in the face of *any* failure event. Using tools from computational geometry, our algorithms are efficient and provide good guarantees.

I. Introduction

With the advent of "cloud" computing, distributed data storage is fast becoming the go-to solution for providing reliable and fast access to data (see, e.g., [8],and [11]). A major issue to consider when designing protocols for such distributed file storage is ensuring reliability and maintaining availability of data in the presence of hardware failures. Of particular importance are failures in the backbone network employed for communication among the data storage "nodes". Such failures may result in network disconnection and thus, loss of access to large portions of the network. For example, in content delivery networks, communication takes place over the internet and data access must be ensured even when failures occur in the internet backbone. At a lower level (e.g. within data-centers), failures in the local backbone must not result in loss of data access. Ensuring reliability in the event of such failures necessitates the presence of an effective data redundancy scheme.

In current systems, there exist two main redundancy schemes for ensuring these requirements: (i) replication of data, where multiple copies of the data are stored (preferably in diverse locations), and (ii) erasure coding, where a set of data fragments is generated (whose total size is larger than the actual data size) and distributed across the network so that we may recover the original data even if a few of these fragments are lost or "erased". These redundancy schemes are usually deployed in a fairly ad-hoc manner by trying to distribute data as widely as possible in the network. However, network failure scenarios often *correlated* as they are often caused by external events such as natural disasters, human error or malicious attacks. Therefore, it is important to study methods of exploiting this correlation in a principled manner to ensure reliability and availability of data. Further, data is often being modified/updated very frequently and maintaining consistency of data is an important requirement. Reconfiguration of the replicas may also be necessary due to efficiency constraints. Thus, it is imperative to reduce communication by minimizing the number of replicas/segments stored in the network while not compromising on reliability and availability.

Geographically Correlated Failures. In this work, we are concerned with *geographically correlated failures*, i.e., where a set of network components fail due to some external events whose effects are geographically correlated. Geographically correlated failures have recently gained attention in the study of network survivability (see [1], [2], [4], [5], [28]–[30], [33]). These types of failures may arise at a higher scale due to the occurrence of natural disasters or even at a lower scale (e.g., within a data-center) due to events such as fires. Figure 1 shows the effects of a geographically correlated failure on the fiber backbone of a major network provider [24]. We model the shapes of such failure regions by considering them as the union of a set of disks/lines in the plane. This models a variety of natural disasters such as earthquakes, hurricanes and solar flares [35] or man-made ones such as Electromagnetic Pulse (EMP) attacks [35], [37], fires or dragging anchors [36].

Network and Redundancy Models. We model the backbone network as a connected graph $G = (V, E)$ em-

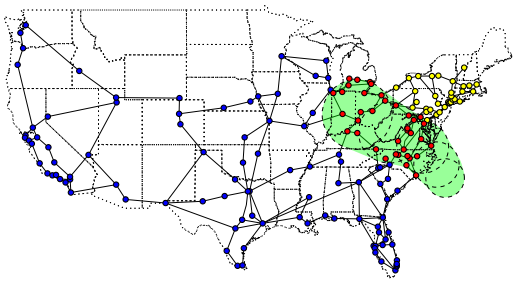


Fig. 1. Impact of a disaster on the fiber backbone of a major service provider [24]. Links in the region shaded green assumed to fail. Yellow nodes are effected directly, but will be disconnected from the blue nodes.

bedded in the plane where V is a set of n node locations (corresponding to routers/switches) and E is a set of m links connecting the nodes (modeled as linear or piecewise linear approximations of the curves in the plane corresponding to the cables running between the routers). In many networks, such as fiber-optic backbones of Internet Service Providers (ISPs) or even smaller-scale backbone networks such as those in data-centers, the graph is "almost" planar (see Figure 1), i.e., only few of the cables cross each other, and thus we assume that G is a planar graph. Individual data servers are connected to the routers and when we refer to storing/placing data at a node, it implies that the data is actually stored at one of the servers connected to it.

In this model, a failure event destroys (permanently, or for periods of hours) some nodes and links from G confined to the shape of the failure events. As a result, the ***residual network*** (surviving network after the failure event) may consist of several maximal connected components. It is worth mentioning that even though some communication via different routes is possible, these routes are likely to be saturated and unstable, hence, in practice, components are considered to be disconnected.

As mentioned above, redundancy may be achieved in one of two ways: ***replication*** and ***erasure coding*** [13]. Replication is the process of placing copies of a data file in multiple nodes such that given a disaster, every surviving node has access to at least one copy. Erasure coding aims to reduce the storage requirement of replication by computing some encoding of the data file and storing small portions of it in different locations so that a node may reconstruct the entire file by retrieving a few of these encoded portions. Formally, a (k', k)-*erasure code* (where $k' \geq k$) splits the data file f into k segments, encodes these k segments in some manner to obtain a set of k' segments to be stored in the network. In this storage scheme the original file can be reconstructed in any node in the network by retrieving and decoding any k of k' segments. Some types of erasure codes such as Fountain codes are "rateless", i.e., the number k' need not be fixed and we can generate as many coded segments as needed.

In what follows, we assume in the replica model that each node can store at most one replica, whereas in the erasure-code model we do not have such a restriction.

Our results. Given a network G modeled as above, a redundancy scheme to be used and a file f, we consider the following file distribution problem: Compute a smallest set of nodes V_D so that, if f is replicated at all nodes in V_D, in the event of *any* geographically correlated failure scenario, f may be recovered at all surviving nodes. That is, if the failure event disconnects the network into a few connected components, each surviving component must contain the entirety of the file so that any node in that component may recover it. We are also interested in the scenario where only those nodes that lie in sufficiently large components have to be recovered, as sometimes the small components may be considered as negligible.

We study the case where failure events are modeled as disks (of either unit or arbitrary radii) or lines in the plane. Specifically, in these cases the disastrous region can lie anywhere in the plane. In the first case this implies that the set of failures Γ is modeled as a continuous set of all translations of a fixed disk, and in the latter case this implies that this set is the continuous set of all translations and rotations of a fixed line. Our main results are the following:

Theorem I.1 (Replication)**.** *Given a planar graph $G = (V, E)$ representing a backbone network, and a geographic correlated failure modeled as a continuous set Γ of disks or lines (as defined above), one can compute a subset of replica nodes $V_D \subseteq V$ such that each connected component in the residual graph G_γ, obtained from G and a failure $\gamma \in \Gamma$, contains at least one node of V_D. The size of V_D is only within $O(\log k^*)$ from the optimum, where k^* is the minimum number of replicas required. The expected running time of the algorithm to construct V_D is $O(k^* n^2 \log^3 n)$ if Γ consists of unit disks or lines, and $O(k^* n^3 \log^3 n)$ if Γ consists of arbitrary radius disks.*

We note that k^* is likely to be much smaller than n, and thus our approximation algorithms "beat" the standard greedy algorithm, which achieves just a $O(\log n)$ guarantee.

As a special case of the problem above, where we are only interested in recovering the large connected components in the residual graph, we obtain:

Corollary I.1. *The data recovery for all connected components of size at least $n\varepsilon$ in the residual network G_γ, for any $\gamma \in \Gamma$, can be completed with only $O(1/\varepsilon \log(1/\varepsilon))$ replicas (obtained as a random sample of the nodes of G). This replica set can be constructed in expected $O(n^2 \log^2 n)$ time if Γ consists of unit disks or lines, and in $O(n^3 \log^2 n)$ expected time for arbitrary-radius disks.*

We also show that our scheme are applicable when disasters have more general shapes. If the shape of the disasters can be approximated as the union of b disks and/or lines, for some integer parameter $b > 0$, then we obtain:

Theorem I.2. *Given the setting of Theorem I.1, and suppose that a geographic correlated failure modeled as a collection of unions $\bigcup_{i=1}^{b} \gamma_i$, for any b-tuple of disks/lines $\gamma_1, \ldots, \gamma_b \in \Gamma$, one can approximate up to a $O(b \log k^*)$ factor the smallest set of nodes $V_D \subseteq V$, so that each connected component in the residual network $G_{\bigcup_{i=1}^{b} \gamma_i}$ contains a vertex of V_D. The expected running time of the algorithm is $n^{O(b)}$.*

Finally, when applying an erasure code, we show:

Theorem I.3 (Erasure coding)**.** *Given the setting of Theorem I.1 and suppose that a (k', k)-erasure code is applied, one can distribute $k' = O(k^* \log k^*)$ encoded segments through the network and guarantee, for each component of size at least k in the residual graph G_γ (for each $\gamma \in \Gamma$), an access to at least k encoded segments, where k^* is the minimum number of segments required to provide the same guarantee. The expected running time is polynomial in n (where Γ consists of arbitrary disks or lines).*

Theorem I.3 implies that employing a (k', k)-coding scheme or a rateless coding scheme to generate k' encoded segments would guarantee recovery of f at every node with sufficient connectivity.

Related Work. Data replication is used in many large-scale distributed storage systems such as Google's BigTable [8] and Amazon's Dynamo [11]. While traditionally, the design of replication schemes has focused on ensuring consistency, such large-scale systems are prone to failures and hence, focus on availability of data as well. The advantage of replication schemes is that they are simple and easier to implement than coding schemes while the disadvantages lie in the storage requirements. Error-correcting codes have been utilized in distributed storage systems such as RAID [32] and DPSS [25] to ensure data availability while being efficient. The key drawback of all of these systems is that they do not explicitly consider the network topology and geographic correlation of failures into account.

The question of how to recover data at a node has been studied in the context of bandwidth requirements [12], [13]. These works consider the tradeoff between bandwidth and storage and focus on data recovery at a single node. Works that consider network topology to design redundancy schemes [21], [27] focus on efficiency of data recovery when data loss occurs while the network is intact. In this, our work differs from these by considering the causes of such data loss directly to decide where to store redundant information.

Whereas the focus of some of the previous works in geographic correlated failures is on (i) assessing the vulnerability of (geographical) networks to such disasters [1], [29], [30] (e.g., identifying disaster locations that have the maximum effect on network capacity), and the impact of such failures [29], (ii) analyzing network vulnerability or evaluate its robustness [2], [4], and measuring its reliability [28], we propose an actual scheme to recover such networks and achieve data reliability in the face of *any* geographic correlated disaster (modeled as a union of disks and lines). To the best of our knowledge, the only previous study which considers the problem of designing data distribution schemes in order to achieve tolerance to geographically correlated failures is the recent work of Banerjee *et al.* [5]. Their work devises a scheme to distribute a set of coded segments generated from the data using an erasure coding scheme into the network, where each node in the network has a storage capacity for storing the file segments. Recovery of the data is only guaranteed in the largest surviving component of the network. They provide an Integer Linear Program (ILP) for this problem and present a $O(\log n)$-approximation algorithm achievable in $O(n^6)$ time, where n is the number of nodes in the network. Our work is somewhat different in the following sense. First, we guarantee a recovery in *every* connected component of the residual network rather just the largest surviving component. When applying an erasure code scheme, our algorithm does not assume storage capacity on the nodes on one hand, but, on the other hand, using techniques from geometric optimization and learning theory we manage to reduce the approximation factor to $O(\log k^*)$ in polynomial time, where k^* is the size of the optimal solution (as above), and the overall majority of the running time is derived by producing a solution to a linear programming system on n variables. In various real-life settings we expect k^* to be much smaller than n. We also propose a solution when one only uses replications to recover data. In this case we show once again a $O(\log k^*)$-approximation factor, where the

running time is $O^*(k^*n^2)$ for unit disk disaster or line disaster (where $O^*(\cdot)$ hides a polylogarithmic factor), and $O^*(k^*n^3)$ for arbitrary-radius disaster. Thus we obtain better guaranteed performance bounds than those in [5] (although the model in [5] is slightly different than ours) and sometimes suggest faster algorithms (at least under the replication model).

We begin with preliminaries in Section II and then present our algorithms and analysis in Section III.

II. Preliminaries

In this section we describe the various steps of our algorithm, so that given a network $G = (V, E)$, modeled as a planar graph, and a geographic correlated failure modeled as a family Γ of translations and rotations of a planar region γ, compute a smallest set of nodes $V_D \subseteq V$, so that each connected component in the residual network G_γ (as defined in Section I), for any $\gamma \in \Gamma$, contains a vertex of V_D. We present solutions for the case where γ is either a disk (of arbitrary radius) or a straight line, where we consider all possible translation and rotations of γ in Γ. In the sequel we show how to formulate our problem as a "hitting-set" problem, and use computational geometry tools in order to achieve a small approximation factor.

Range spaces. We begin by reviewing a few concepts from random sampling and statistical learning theory [31]. Let X be a (possibly infinite) set of objects also referred to as the "space", and let $\mathcal{R}$ be a (possibly infinite) family of subsets of X, those subsets are called the "ranges". The pair $\Sigma = (X, \mathcal{R})$ is called a *range space*.

In our context, we define the range space $\mathcal{H} = (V, \mathcal{C}_\Gamma)$, where V is the vertex set of G as above, and $\mathcal{C}_\Gamma$ consists of the collection of all subsets of V that appear in a connected component of the residual graph $\mathcal{G}_\gamma$ (that is, after excluding from G all nodes and links meeting γ), for each $\gamma \in \Gamma$. Although at first glance it may appear that $\mathcal{C}_\Gamma$ is a very large collection (e.g., it may contain exponentially many connected components), a main property shown in Theorem III.1 is the fact that $|\mathcal{C}_\Gamma|$ is only *polynomial* in $|V|$ (and this property holds for any restriction of $\mathcal{C}_\Gamma$ to a subset of V)—see below.

ε-nets. Given a range space $(X, \mathcal{R})$ as above and a parameter $\varepsilon > 0$, a subset $N \subseteq X$ is called an *ε-net* of $(X, \mathcal{R})$ if $R \cap N \neq \emptyset$ for all $R \in \mathcal{R}$ such that $|R \cap X| \geq \varepsilon |X|$. In other words, an ε-net is a hitting set for all the "heavy" ranges. A seminal result by Haussler and Welzl [20] shows that under some favorable scenarios (including various geometric settings) a random subset $N \subseteq X$ of size $O(1/\varepsilon \log(1/\varepsilon))$ is an ε-net of Σ with constant probability. The notion of ε-nets can be extended to the case where the elements in X are assigned weights. In this case a *weighted ε-net* is a subset $N \subseteq X$ that hits all ranges $R \in \mathcal{R}$ whose total weight (that is, the sum of the weights over the elements $x \in R$) is at least ε of the total weight of X.

Hitting sets. A *hitting set* for a range space $(X, \mathcal{R})$ is a subset $H \subseteq X$ such that each range $R \in \mathcal{R}$ intersects H. The *hitting-set problem* is to find a smallest-size subset $H \subseteq X$ with this property. A related (dual) problem is the *set-cover problem* (or *coverage problem* as referred in sensor networking literature) is to find a smallest subcollection of ranges $\mathcal{S} \subseteq \mathcal{R}$ that altogether cover X. The general (primal and dual) problems are NP-Hard to solve (even approximately) [17], [22], and the simple greedy algorithm yields the (asymptotically) best known approximation factor of $O(\log n)$ computable by a polynomial-time algorithm [6], [15]. Most of these problems remain NP-Hard even in geometric settings [14], [16]. However, one can attain an improved approximation factor of $O(\log k^*)$ in polynomial time for many of these scenarios, where k^* is the size of the optimal solution. This improvement is based on the iterative reweighting scheme of Brönnimann and Goodrich [7] (see also Clarkson [10]), which is strongly related to ε-nets. We briefly review this technique below.

Iterative reweighting scheme. We are given a range space $(X, \mathcal{R})$, and the goal is to find a small subset H of X which meets every range in $\mathcal{R}$. The technique assumes the availability of two black-box routines: (i) an *ε-net finder*, which is a procedure that, given any weight function w on X and $\varepsilon > 0$, constructs a *weighted* ε-net N for $(X, \mathcal{R})$, in the sense that N hits each range whose weight is at least $\varepsilon w(X)$; (ii) a *verifier*, which is a procedure that, given a subset $H \subseteq X$, either determines that H is a hitting set, or returns a nonempty range $R \in \mathcal{R}$ such that $R \cap H = \emptyset$.

The algorithm runs an exponential search to guess the value of k^*, which is the size of the smallest hitting set for $(X, \mathcal{R})$. At each step, denote by k^* the current guess for this value. The algorithm assigns weights (initially, uniform) to the elements of X, and uses the net finder to select a (weighted) $\frac{1}{2k^*}$-net N. If the verifier determines that N is a hitting set, it outputs N and stops. Otherwise, it returns some range R not hit by N. We double the weights of the elements in R, and repeat the above procedure. We keep iterating in this manner until N hits all the ranges in $\mathcal{R}$, or abort after a pre-specified number of iterations, concluding then (with high probability) that the current guess for k^* is too small. Thus, upon termination, the size of the reported hitting set has the same upper bound as that for $\frac{1}{2k^*}$-

nets. If we are in the scenario where this bound is $O(k^* \log k^*)$ (as mentioned above), then the resulting approximation factor is $O(\log k^*)$.

The analysis of [7] (see also [10]) shows that, at the right guess for k^*, the algorithm terminates after at most $O(k^* \log (n/k^*))$ rounds. Thus the overall running time of the algorithm is

$$O\left(k^* \log (n/k^*)(T_N + T_V)\right), \tag{1}$$

where T_N, T_V are the respective running time bounds for the net-finder and the verifier.

Our problem. Our problem of computing a smallest set of nodes $V_D \subseteq V$, so that each connected component in the residual network contains a vertex of V_D is nothing but a hitting-set problem. Indeed, recall that our range space $\mathcal{H} = (V, \mathcal{C}_\Gamma)$ consists of the vertices V of G and the collection of all subsets of V that appear in a connected component of the residual graph. Thus our actual goal is to compute a smallest subset of $V_D \subseteq V$ that hits each range in $\mathcal{C}_\Gamma$ (and thus each connected component in the residual graph).

III. A Hitting-Set Based Algorithm for Replica Placement

Our algorithm exploits the iterative reweighting scheme reviewed above. To this end, we first describe the specification of an ε-net and a verifier in the context of our problem, and then describe the algorithms to complete these two procedures. We show below that one can (i) construct in expected polynomial time an ε-net of size $O(1/\varepsilon \log (1/\varepsilon))$, and (ii) implement the verifier procedure in polynomial time. Having this machinery at hand, we plug it into the iterative reweighting scheme reviewed above, and obtain a $O(\log k^*)$-approximation algorithm for our hitting-set problem, with expected polynomial running time.

A. An ε-net Finder

In our setting, an ε-net for the range space $(V, \mathcal{C}_\Gamma)$ is a subset $N \subseteq V$ that hits each connected component of size at least εn in the residual graph G_γ, for any $\gamma \in \Gamma$.

The main challenge that we face in this part of the analysis is *combinatorial*, that is, we aim to show that our range space $(V, \mathcal{C}_\Gamma)$ indeed admits an ε-net of size $O(1/\varepsilon \log (1/\varepsilon))$. To this end, we present the notion of *VC-dimension*.

VC-dimension. Given a range space $\Sigma = (X, \mathcal{R})$, a finite subset $Y \subseteq X$ is *shattered* by Σ if $\{R \cap Y \mid R \in \mathcal{R}\} = 2^Y$, i.e., the restriction of $\mathcal{R}$ to Y can realize all subsets of Y. The *VC-dimension* of Σ is defined to be the largest size of a subset of X shattered by Σ; the VC-dimension is infinite if the size of the shattered subset is arbitrarily large.

Many natural range spaces have finite VC-dimension. A main property of range spaces $(X, \mathcal{R})$ with VC-dimension d is the fact that the number of their ranges is only $O(n^d)$ [31] (that is, it is polynomial in n rather than exponential without this assumption), and this property continues to hold in any restriction of $\mathcal{R}$ to $Y \subseteq X$. In fact, in various applications it is sufficient to have only this assumption about the number of ranges (from which a finite VC-dimension follows), nevertheless, our analysis is directly related to range spaces of finite VC-dimension, which is the reason we describe this notion in detail. Returning to our discussion about ε-nets, the bound on their size, shown in [20], is in fact $O((d/\varepsilon)\log(d/\varepsilon))$. Specifically, a random sample of that size is an ε-net of Σ with constant probability. This bound has later been improved to $O(\frac{d}{\varepsilon} \log \frac{1}{\varepsilon})$ by Komlós, Pach, and Woeginger [23], who showed that a random sample of that size is an ε-net with constant probability.

Our setting. In order to show that our setting admits an ε-net of size $O(1/\varepsilon \log (1/\varepsilon))$, we show that the VC-dimension d of $(V, \mathcal{C}_\Gamma)$ is finite. Then, we construct such an ε-net by randomly sampling $O(1/\varepsilon \log (1/\varepsilon))$ vertices of V (with a constant of proportionality that depends on d) and obtain an appropriate sample after a constant number of trials on average. The construction of a weighted ε-net is somewhat more involved but can be done in $O(n)$ time using standard machinery[1], see, e.g., [26]. Thus the actual ε-net construction is standard, while the bound on its size requires a considerably more involved analysis. We thus devote the remainder of the analysis concerning the net-finder to show that $(V, \mathcal{C}_\Gamma)$ has a finite VC-dimension, which is proved in Theorem III.1.

We note that in other (simpler) geometric settings given by a range space $(X, \mathcal{R})$ of, say, points and simply-shaped regions, it is almost immediate to show that the number of ranges in any restriction of $(X, \mathcal{R})$ to a smaller subset $X' \subseteq X$ is polynomial in $|X'|$. For example, for points and halfspaces in d-space (for d constant), we always have $O(n^d)$ halfspaces defined over any subset of n points (as each hyperplane bounding a halfspace is defined by d points), for points and balls in d-space this bound becomes $O(n^{d+1})$ (as each ball is defined by at most $d+1$ points), and there are various

[1]Roughly speaking, each vertex of V is "duplicated" according to its weight, and then, after an appropriate weight normalization, a random sample of $O(1/\varepsilon \log (1/\varepsilon))$ elements from this new setting is a weighted ε-net with constant probability

examples of this kind. Such considerations immediately show that the VC-dimension is finite.

Nevertheless, a major difficulty in our setting $(V, \mathcal{C}_\Gamma)$ is the fact that the graph G does not have a constant description complexity. Specifically, a connected component in $\mathcal{C}_\Gamma$ is not necessarily defined by a constant number of vertices (that is, we no longer have the property that a range is defined by a constant number of elements from the space). Moreover, by definition of restricted range spaces, in the restriction of $(V, \mathcal{C}_\Gamma)$ to a subset $V' \subseteq V$, we do not modify the paths connecting vertices in the original graph G, but only consider the connectivity among the vertices in V' (w.r.t. the original graph). Thus each connected component $C \in \mathcal{C}_\Gamma$ is now confined to V'. Given this setting, the problem of showing that the number of such restricted ranges is only polynomial in $|V'|$ becomes non-trivial. In fact, our verifier procedure reviewed in Section III-B shows that this number is polynomial when $V' = V$ (that is, in the original range space), but it does not necessarily apply to any subset of V. This is the main reason we need to resort to the analysis in this section. Our analysis is intricate and involves tools from combinatorial geometry, and thus in this version of the paper we only provide a sketch of our proof, and defer the rest of the analysis details to the full version of this paper.

Theorem III.1. *The range space $(V, \mathcal{C}_\Gamma)$ has VC-dimension 4, when Γ is either a set of disks or lines in the plane.*

Proof sketch: We only present the case where Γ consists of disks. The case of lines is addressed in the full version of this paper, and is an easy generalization for the case of disks. In fact, for the case of disks our analysis does not exploit their geometry but only their combinatorial structure. That is, it only exploits the fact that for any pair of disks their boundaries are either disjoint or meet exactly twice; these regions are referred to as *pseudo-disks*.

Recall that we assume our graph G is planar and connected. Our goal is to show that one cannot shatter a subset of vertices of size 5. Meaning that given any subset $U \subseteq V$ of five vertices, one cannot obtain all 2^U subsets appearing as connected components in the residual graph G_γ, for any $\gamma \in \Gamma$. In this case, we assume by contradiction that this shattering is possible, in particular, this implies that all pairs of vertices $\{u, u'\}$ in U are realizable, in which case there is always a region $\gamma \in \Gamma$ such that (i) it leaves u, u' and the path (in G) connecting them intact, and (ii) intersects all other paths connecting the remaining pairs in U (and so now these pairs are disconnected). Consider now all these pairs $\{u, u'\}$ and the paths $P_{u,u'}$ connecting them, they form a "clique graph" of five vertices (with edges replaced by these paths). It is well known from the theory of combinatorial geometry that this clique cannot be embedded in the plane so that all its edges are crossing-free (see, .e.g., [18]). Then, using a theorem of Hannani and Tutte [18], [34], it implies that there must be at least one such pair of paths $P_{u,u'}$, $P_{v,v'}$ that meet an odd number of times (as this theorem implies that when each pair of paths cross only an even number of times, one can draw the corresponding graph in the plane without any crossing). Then, using the fact that G is planar and that Γ is a set of pseudodisks, we obtain that $P_{u,u'}$, $P_{v,v'}$ meet an even number of times, which is a contradiction to our assumption. We thus conclude that at least one pair in U cannot be realized, and thus a subset of size five cannot be shattered. Thus one can shatter at most four vertices. □

B. The Verifier

Our verification process proceeds as follows. Given a planar graph $G = (V, E)$ modeling our network, a set Γ of geographic correlated failures modeled as a collection of planar regions (each of which is a translation or rotation of a fixed planar region γ), and a set $V_D \subset V$ of replicas, determines whether V_D hits each connected component in the residual graph G_γ, for any $\gamma \in \Gamma$. A "YES" answer implies that the given set of replicas ensures a recovery of G in the face of any geographic correlated failure (from Γ). A "NO" answer implies that there is a connected component in the residual graph G_γ, for some $\gamma \in \Gamma$ that is missed by V_D. In this case we return this connected component.

Our verifier procedure is based on the algorithm of Agarwal *et al.* [2]. We thus only briefly describe our approach, then apply the algorithm in [2], and then state its running time under the cases where Γ is either a set of disks or lines in the plane.

In each of these cases, Γ consists of a continuous collection of regions (each of which is a translation and rotation of a fixed region γ), nevertheless, it is only sufficient to consider a set of special events, referred to as *critical events* in which the structure of the residual graph G_γ changes, and then new connected components may be created (as a split of existing components), or existing components may be merged together. Only in the events of the former kind the (given) set V_D may miss a connected component (just created), in which case we return a "NO" answer. The "merging" events must be tracked as well, in order to be able to report all connected components of the residual graph at any time

(as this components may be split later on). The machinery given in [2] enables us to efficiently track all these events. Roughly speaking, the algorithm in [2] performs a walk on some planar subdivision induced by regions corresponding to the edges e of G and the objects in Γ. Each point on this planar subdivision corresponds to a specific location of γ, and each region in this subdivision corresponds to all locations of γ in which it meets an edge e of G, for each e. Each cell of this subdivision corresponds to placements of γ where the residual graph G_γ (in each of these locations) remains unchanged. The transition between adjacent cells corresponds to the critical events, as in each transition γ meets a new edge e of G or becomes apart from an edge e' of G. The running time of the algorithm in [2] is proportional to the complexity of this subdivision, in fact, it adds only a multiplicative factor of $O(\log^2 n)$ over the subdivision complexity. Thus we only need to show these bounds in our scenario in order to bound the running time of the verifier procedure. We also note that the mechanism in [2] enables us to test for each newly created connected component whether it contains a vertex of V_D, and make a counting query on its size (i.e., number of vertices it contains) within the same asymptotic running time.

We thus only provide a brief description of the resulting subdivision when γ is either a disk or a line, and then state its complexity. We use a known fact from the theory of "arrangements" (see, e.g., [3]) that subdivisions of simply-shaped regions in d-space have only $O(n^d)$ cells.

First, when γ is a unit disk, then, as observed in [2], the regions comprising our subdivision are *hippodromes*. Specifically, each of these regions corresponds to the *Minkowski Sum* of an edge $e \in E$ and a unit disk (see, e.g., [31]), that is, this region is $\{x \in \mathbb{R}^2 \mid C(x) \cap e \neq \emptyset\}$, where $C(p)$ is the unit disk centered at p. In our case, we have $O(n)$ hippodromes, as there are only $O(n)$ edges in G. In this case the planar subdivision will contain only $O(n^2)$ cells, and thus the verifier runs in $O(n^2 \log^2 n)$ time. See Figure 2(a) for an example.

When γ is a disk of an arbitrary radius, we proceed as follows. Let r be the radius of γ and let (x, y) be its center. We now represent γ by a point $(x, y, r) \in \mathbb{R}^3$, and thus obtain a three-dimensional subdivision of simply-shaped regions with only $O(n^3)$ cells, from which it will follow that the verifier runs in $O(n^3 \log^2 n)$ time. Specifically, for each link $e \in E$, we consider the bivariate function $f_e : \mathbb{R}^2 \to \mathbb{R}$ defined by $f_e(p) = d(p, e)$ where $d(p, e)$ is the distance from p to the closest point on e. The region thus created is the Minkowski Sums of the edge $e \in E$ and a *cone*, and, by definition, each point (x, y, r) inside this region represents a disk of radius r centered at (x, y) which intersects e. In particular, the hippodrome of e defined for unit disks is obtained at the level set $\{p \in \mathbb{R}^2 \mid f_e(p) = 1\}$.

When γ is a line in the plane, we apply our analysis in the *dual plane* in order to classify all locations and rotations of γ for which it intersects an edge of G. Roughly speaking, a standard duality transform (see, e.g., [31]) maps points to lines and lines to points. In particular, this implies that a line γ is mapped to a point γ^* and a segment representing an edge e is mapped to a *double wedge* e^* (once again, see [31]). Moreover, γ meets e if and only if the point γ^* lies in the double wedge e^*. We now form the planar subdivision of these double wedges in order to track all critical events. The overall number of cells we obtain is $O(n^2)$, and thus the verifier runs in $O(n^2 \log^2 n)$ time. See Figure 2(b)–(c) for an illustration.

Summarizing the above discussion, we have shown:

Corollary III.1. *The running time of the verifier procedure is $O(n^2 \log^2 n)$ if γ is a line or a unit disk, or $O(n^3 \log^2 n)$ if γ is an arbitrary-radius disk.*

wrapping up. Recall that the ε-net finder runs in $O(n)$ time. Putting this together with our verifier procedures into the iterative reweighting scheme in (1), we prove Theorem I.1.

As a corollary from Theorem III.1 we would like to address the simpler scenario where we only wish to recover the data at the large connected components. Let us set a threshold parameter $\varepsilon > 0$, and suppose we would like to recover only those components of size $\geq \varepsilon n$. By our discussion about ε-nets it follows that only $O(1/\varepsilon \log(1/\varepsilon))$ replicas are sufficient. We can produce this replica set by just randomly sample that many elements, and then use the verification procedure to test all large connected components (we follow the same procedure almost verbatim with the same asymptotic running time). Since such a random sample is an ε-net with constant probability [20], we need only a constant number of such trials in average. This shows Corollary I.1.

Real-life settings. We note that in real-life settings the edges in the network G are likely to be "apart" from each other. In the context of unit-disk disasters it implies that the corresponding planar subdivision has only a near-linear complexity rather than quadratic, from which it follows the running time for the verifier is nearly-linear. This speeds up the running time of our hitting-set algorithm to be close to $O(nk^*)$ (up to some polylogarithmic factor). In Figure 3 we list several such settings (see also [1] for these details).

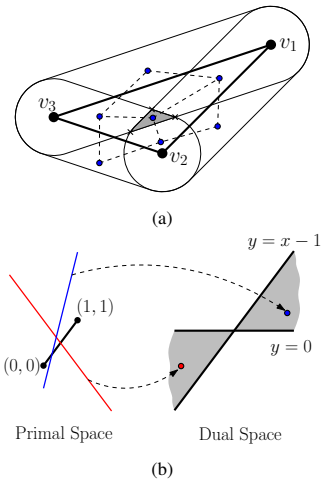


Fig. 2. (a) Hippodromes for three links and their subdivision. A single cell is highlighted. The graph G' is also shown.(b) The point-line duality transformation.

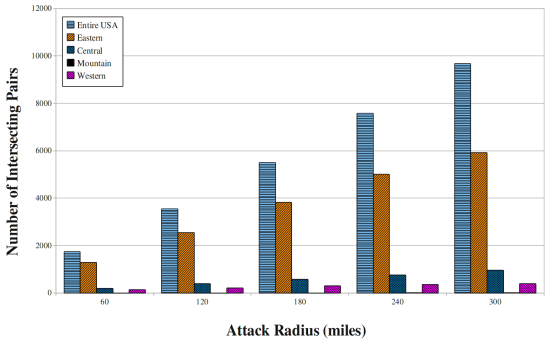


Fig. 3. The number of intersecting pairs of hippodromes for attack radii 60–300 miles. The results are shown for the fiber-optic network presented in Figure 1 and for various sub-networks (determined by different US time-zones).

C. Extension: Modeling failure events of arbitrary shape

We now resolve the case of arbitrary-shape disasters in the replica model, where these shapes are modeled as union of (possibly, arbitrary-radius) disks and/or lines; in a typical setting an arbitrary shape is approximated in this manner.

Specifically, letting Γ be a set of (arbitrary-radius) disks and lines, we now consider the residual graph of G after placing b distinct regions $\gamma_1, \ldots, \gamma_b \in \Gamma$, where $b > 0$ is an integer parameter. We now obtain the range space $(V, \mathcal{C}_\Gamma^{(b)})$ consisting of the nodes V of G and the collection of all subsets of V that appear in a connected component of $G_{\bigcup_{i=1}^b \gamma_i}$ for some b-tuple of regions $\gamma_1, \ldots, \gamma_b \in \Gamma$, where $G_{\bigcup_{i=1}^b \gamma_i}$ is the residual graph after excluding from G all nodes and links meeting $\bigcup_{i=1}^b \gamma_i$.

We now observe that each connected component of $G_{\bigcup_{i=1}^b \gamma_i}$ is obtained by the *intersection* of b connected components (that is, b ranges) from the original range space $(V, \mathcal{C}_\Gamma)$. Indeed, such a connected component is affected by b distinct regions $\gamma_1, \ldots, \gamma_b$, each of which defines a connected component C_{γ_i} in its own (individual) residual graph G_{γ_i}, $i = 1 \ldots, b$. Thus a connected component of $G_{\bigcup_{i=1}^b \gamma_i}$ appears in the intersection $\bigcap_{i=1}^b C_{\gamma_i}$ (note that this intersection may not be connected). Using standard VC-dimension properties and the fact that the original range space $(V, \mathcal{C}_\Gamma)$ has a finite VC-dimension, one can show that the VC-dimension of $(V, \mathcal{C}_\Gamma^{(b)})$ is at most $O(b)$; see, e.g., [19].

ε-net finder and verifier procedures. By the ε-net bound in [23] it follows that $(V, \mathcal{C}_\Gamma^{(b)})$ has an ε-net of size $O\left(\frac{b \log(1/\varepsilon)}{\varepsilon}\right)$ (in fact, a random sample of than size is an ε-net with constant probability).

Concerning the verifier, we are not aware of how to efficiently extend it to the case $b > 1$. Instead, we just resort to a brute-force algorithm, which exhausts all $n^{O(b)}$ ranges in $(V, \mathcal{C}_\Gamma^{(b)})$, and then tests each of them with respect to the candidate hitting set. Omitting any further details, the running time is $n^{O(b)}$ (with a constant of proportionality in the exponent being slightly larger than that in the bound on the number of ranges). Incorporating these procedures into the iterative reweighting scheme we obtain Theorem I.2.

D. Extension: Erasure Coding

In this section, we extend the previous algorithms to the case when erasure coding is used. We are given a file f split into k segments and assume that a (k', k)-coding scheme is used where $k' \geq k$. Our goal is that, after a disastrous event, each surviving node connected to at least k other nodes will have access to at least k segments of f, as it will enable the recovery of the data in each node. We aim to choose a smallest subset of coding nodes that guarantees this property. This is particularly applicable in scenarios where rateless codes are used and as many encoded segments may be generated as necessary.

A reduction to multi-hitting set. Analogously to hitting sets, a ***k-hitting set*** for a range space $(X, \mathcal{R})$ is a set $S \subseteq X$ such that, for every $R \in \mathcal{R}$, $|S \cap R| \geq k$. The MIN-k-HITTING-SET problem is that of computing a minimum cardinality k-hitting set. In order to maintain an accessibility of each surviving node v to at least k other nodes storing segments of f (so that v will be able to fully reconstruct f), we need to solve a MIN-k-HITTING-SET problem for the range space $(V, \mathcal{C}_\Gamma)$.

Indeed, a subset $V_D \subset V$ is a k-hitting set for $(V, \mathcal{C}_\Gamma)$, if each connected component in $\mathcal{C}_\Gamma$ is hit by at least k nodes of V_D.

We derive a solution for this problem by using the machinery of Chekuri *et al.* [9], who obtained an $O(d \log k^*)$-approximation factor (computed in expected polynomial time, given a polynomial-time verifier procedure) for the MIN-k-HITTING-SET problem when d is the VC-dimension. Thus in our scenario, since $d = 4$, we obtain a $O(\log k^*)$-approximation factor.

We note that the algorithm in [9] was originally presented for the set-cover problem (dual to hitting set), and it exploits an LP-relaxation and the fact that range spaces of finite VC-dimension have small "relative approximations" (roughly speaking, these structures emerge from learning theory, generalize ε-nets, and guarantee that the size of each range is approximated up to some relative error). The overall majority of the running time is in producing a solution to a linear programming system (which we then relax). Using the standard Ellipsoid or Interior-point algorithms this can be completed in polynomial time. We have thus shown Theorem I.3.

REFERENCES

[1] P. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman. Network vulnerability to single, multiple, and probabilistic physical attacks. In *Proc. 2010 IEEE MILCOM*, pages 1824–1829, 2010.

[2] P. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman. The resilience of wdm networks to probabilistic geographical failures. In *Proc. 30th IEEE INFOCOM*, pages 1521–1529, 2011.

[3] P. K. Agarwal and M. Sharir. Arrangements and their applications. In *Handbook of Computational Geometry*, chapter 2, pages 49–119. Elsevier, 2000.

[4] S. Banerjee, S. Shirazipourazad, P. Ghosh, and A. Sen. Beyond connectivity - new metrics to evaluate robustness of networks. In *Proc. 12th IEEE HPSR*, pages 171 –177, 2011.

[5] S. Banerjee, S. Shirazipourazad, and A. Sen. On region-based fault tolerant design of distributed file storage in networks. In *Proc. 31st IEEE INFOCOM*, pages 2806 –2810, 2012.

[6] M. Bellare, S. Goldwasser, G. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In *Proc. 25th Annu. ACM Symp. Theory Comput.*, pp. 294–304, 1993.

[7] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite VC dimensions. *Discrete Comput. Geom.*, 14:463–479, 1995.

[8] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2):4:1–4:26, 2008.

[9] C. Chekuri, K. L. Clarkson, and S. Har-Peled. On the set multicover problem in geometric settings. *ACM Trans. Algorithms*, 9(1):9:1–9:17, 2012.

[10] K. L. Clarkson. Algorithms for polytope covering and approximation. In *Proc. 3rd Workshop on Algorithms and Data Structures*, pages 246–252, 1993.

[11] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: amazon's highly available key-value store. *SIGOPS Oper. Syst. Rev.*, 41(6):205–220, 2007.

[12] A. Dimakis, V. Prabhakaran, and K. Ramchandran. Decentralized erasure codes for distributed networked storage. *Information Theory, IEEE Transactions on*, 52(6):2809 – 2816, 2006.

[13] A. Dimakis, K. Ramchandran, Y. Wu, and C. Suh. A survey on network codes for distributed storage. *Proc. IEEE*, 99(3):476 –489, 2011.

[14] T. Feder and D. H. Greene. Optimal algorithms for approximate clustering. In *Proc. 20th Annu. ACM Symp. Theory Comput.*, pages 434–444, 1988.

[15] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[16] R. Fowler, M. Paterson, and S. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Inform. Process. Lett.*, 12(3):133–137, 1981.

[17] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, 1979.

[18] H. Hanani (Chaim Chojnacki). Über wesentlich unpläattbare Kurven im drei-dimensionalen Raume. *Fundamenta Mathematicae*, 23:135–142, 1934.

[19] S. Har-Peled. *Geometric Approximation Algorithms*. Mathematical Surveys and Monographs, Volume 173. American Mathematical Society, 2011.

[20] D. Haussler and E. Welzl. ε-nets and simplex range queries, *Discrete Comput. Geom.*, 2:127–151, 1987.

[21] A. A. Jiang and J. Bruck. Network file storage with graceful performance degradation. *Trans. Storage*, 1(2):171–189, 2005.

[22] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller, J. W. Thatcher, Eds., *Complexity of Computer Computations*, pages 85–103, Plenum Press, New York, 1972.

[23] J. Komlós, J. Pach, and G. Woeginger. Almost tight bounds for epsilon nets. *Discrete Comput. Geom.*, 7:163–173, 1992.

[24] Level 3 Communications. Network map. http://www.level3.com/interacts/map.html.

[25] Q. Malluhi and W. Johnston. Coding for high availability of a distributed-parallel storage system. *IEEE Trans. Parallel Distrib. Syst.*, 9(12):1237 –1252, 1998.

[26] J. Matoušek. Approximations and optimal geometric divide-and-conquer. *J. Comput Sys. Sci.*, 50(2):203–208, 1995.

[27] M. Naor and R. M. Roth. Optimal file sharing in distributed networks. *SIAM J. Comput.*, 24(1):158–183, 1995.

[28] S. Neumayer and E. Modiano. Network reliability with geographically correlated failures. In *Proc. IEEE INFOCOM*, Mar. 2010.

[29] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano. Assessing the impact of geographically correlated network failures. In *Proc. IEEE MILCOM'08*, 2008.

[30] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano. Assessing the vulnerability of the fiber infrastructure to disasters. In *Proc. IEEE INFOCOM'09*, 2009.

[31] J. Pach and P. Agarwal. *Combinatorial Geometry*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011.

[32] D. A. Patterson, G. Gibson, and R. H. Katz. A case for redundant arrays of inexpensive disks (raid). In *Proc. 1988 ACM SIGMOD*, pages 109–116, 1988.

[33] A. Sen, S. Murthy, and S. Banerjee. Region-based connectivity: a new paradigm for design of fault-tolerant networks. In *Proc. IEEE HPSR*, 2009.

[34] W. Tutte. Toward a theory of crossing numbers. *J. Comb. Theory*, 8(1):45–53, 1970.

[35] U.S. Federal Energy Regulatory Commission, Dept. of Homeland Security, and Dept. of Energy. Detailed technical report on EMP and severe solar flare threats to the U.S. power grid. Oct. 2010.

[36] R. Wilhelm and C. Buckridge. Mediterranean fibre cable cut – a RIPE NCC analysis. Apr. 2008.
[37] C. Wilson. High altitude electromagnetic pulse (HEMP) and high power microwave (HPM) devices: Threat assessments. CRS Report for Congress, July 2008.

APPENDIX

Assume to the contrary that there are five vertices $v_1, \ldots, v_5$ of G such that $\{v_1, \ldots, v_5\}$ is shattered by $H(G, \mathcal{C})$. In particular this means that for every $1 \leq i < j \leq 5$ there is $C_{ij} \in \mathcal{C}$ such that v_i and v_j belong to the same connected component of $G_{C_{ij}}$ and this connected component does not contain any of $\{v_1, \ldots, v_5\} \setminus \{v_i, v_j\}$. Hence for every $1 \leq i < j \leq 5$ there exists a simple path P_{ij} in $G_{C_{ij}}$ connecting v_i and v_j. Observe that, by definition, P_{ij} does not meet the curve C_{ij}.

In order to handle more elegantly the possible case in which two paths P_{ij} and $P_{k\ell}$ share common edges and vertices other than their endpoints, we do the following: For every $1 \leq i < j \leq 5$ we draw a curve Q_{ij} connecting v_i and v_j such that Q_{ij} follows very closely the path P_{ij}. We make sure that Q_{ij} and P_{ij} meet exactly the same set of curves in $\mathcal{C}$. The difference between P_{ij} and Q_{ij} is that Q_{ij} is drawn in a rather free fashion and it does not meet any edge of G more than at a finite number of points (see Figure 4 for an illustration). We will also make sure that at every point that two curves Q_{ij} and $Q_{k\ell}$ meet they either properly cross or share a common endpoint.

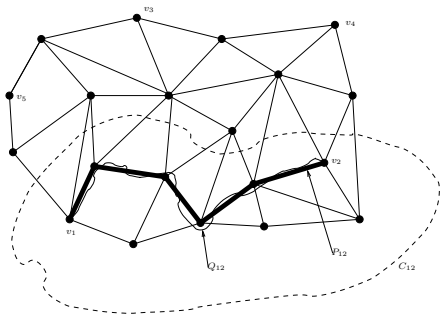
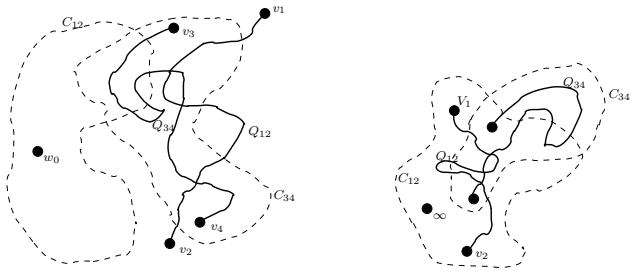


Fig. 4. The path P_{12} and the curve Q_{12}.

Considering only the curves Q_{ij} for $1 \leq i < j \leq 5$ and the vertices $v_1, \ldots, v_5$, we obtain a drawing of K_5 in the plane. As K_5 is not a planar graph, by a theorem of Hannani and Tutte [18], [34], there exist four distinct indices, that without loss of generality we assume they are $1, 2, 3$, and 4, such that Q_{12} and Q_{34} cross an odd number of times.

Recall that the curve Q_{12} is disjoint from C_{12}. We may assume, without loss of generality, that Q_{12} is contained in the area bounded by the simple closed curve C_{12} rather than in its complement. Indeed, if it happens that Q_{12} is contained in the region that is not bounded by C_{12}, we can apply inversion mapping with respect to a point w_0 that lie in the area bounded by C_{12} (specifically, this mapping takes a point z in the complex plane to the point $\frac{1}{z-w_0}$). Since the inversion mapping is continuous it does not effect any incidence relation and the global combinatorial picture of our curves remains the same. However, after applying the inversion mapping the curve Q_{12} is now contained in the region bounded by C_{12} (see Figure 5 for an illustration).



(a) Before applying inversion

(b) After applying inversion

Fig. 5. Applying the inversion mapping.

Let x_1 be the first intersection point of Q_{12} with Q_{34} as we traverse Q_{12} from v_1 in the direction of v_2. Let x_2 be the first intersection point of Q_{12} with Q_{34} as we traverse Q_{12} from v_2 in the direction of v_1 (it may happen that $x_1 = x_2$, in which case Q_{12} and Q_{34} meet only at $x_1 = x_2$). Similarly, let x_3 be the first intersection point of Q_{34} with Q_{12} as we traverse Q_{34} from v_3 in the direction of v_4. Let x_4 be the first intersection point of Q_{34} with Q_{12} as we traverse Q_{34} from v_4 in the direction of v_3 (see Figure 6).

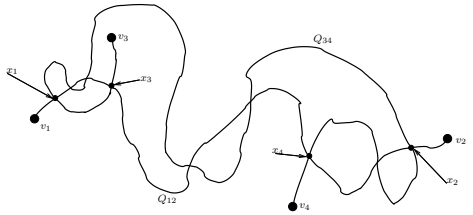


Fig. 6. The definition of x_1, x_2, x_3, and x_4.

For convenience we introduce the following notation. Given a curve Q and two points a and b on Q we denote by $Q(a, b)$ the portion of Q delimited by a and b (we

assume Q is not a closed curve).

We note the following simple yet important observation.

Claim A.1. 1) *The curve C_{34} must cross $Q_{12}(v_1, x_1)$.*
2) *The curve C_{34} must cross $Q_{12}(v_2, x_2)$.*
3) *The curve C_{12} must cross $Q_{34}(v_3, x_3)$.*
4) *The curve C_{12} must cross $Q_{34}(v_4, x_4)$.*

Proof. Because the four statements are very similar we prove only the first one leaving the rest to the reader. Assume to the contrary that C_{34} does not cross $Q_{12}(v_1, x_1)$. As we traverse the path P_{12} from v_1 in the direction of v_2, let v be the first vertex on P_{12} that is also a vertex of P_{34}. Such a vertex must exist because P_{12} and P_{34} meet as Q_{12} and Q_{34} cross each other. Because G is a planar map P_{12} and P_{34} must first meet at a vertex of G. Now it is easy to see that, because C_{34} does not cross $Q_{12}(v_1, x_1)$, v_1 belongs to the same connected component of $G_{C_{34}}$ as v_3 and v_3, contradicting the definition of C_{34}. □

Consider the curve Q_{12}. As we traverse this curve from v_1 in the direction of v_2 let $z_1, \ldots, z_k$ be the intersection points of Q_{12} and C_{34} listed in the order they are encountered.

We claim that $Q_{12}(v_1, z_1)$ does not meet the curve Q_{34}. Indeed, this follows from Claim A.1 because if $Q_{12}(v_1, z_1)$ meets Q_{34} at a point w, then the point x_1 must lie on $Q_{12}(v_1, w)$. In particular $Q_{12}(v_1, x_1)$ does not meet the curve Q_{34}, as z_1 is the first meeting point of Q_{12} and Q_{34} as we traverse Q_{12} from v_1 in the direction of v_2. This is a contradiction to Claim A.1. In a completely analogous manner one can show that $Q_{12}(v_2, z_k)$ does not meet the curve Q_{34}.

We will reach a contradiction by showing that each of the portions $Q_{12}(z_i, z_{i+1})$ meets Q_{34} an even number of times. Indeed, this will contradict the assumption that Q_{12} and Q_{34} meet an odd number of times.

Let $1 \leq i < k$ and consider the portion $Q_{12}(z_i, z_{i+1})$ of Q_{12}. Recall that z_i and z_{i+1} are two points on C_{34}. The points z_i and z_{i+1} divide C_{34} into two subarcs that we denote by A and B. Observe that the union of A and $Q_{12}(z_i, z_{i+1})$ is a simple closed curve that we denote by W_A (see Figure 7). Similarly, the union of B and $Q_{12}(z_i, z_{i+1})$ is a simple closed curve that we denote by W_B.

Because every two simple closed curves cross each other an even number of times, C_{12} crosses W_A an even number of times. It follows that C_{12} crosses A an even number of times, as C_{12} is disjoint from $Q_{12}(z_i, z_{i+1})$. Similarly, C_{12} crosses B an even number of times.

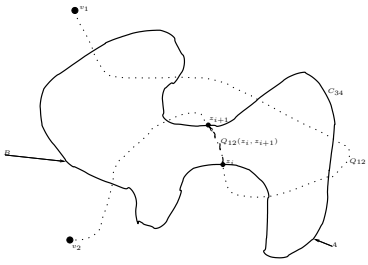


Fig. 7. The definition of A and B.

However, because $\mathcal{C}$ is a family of pseudo-circles C_{12} and C_{34} may cross each other at most twice. This implies that C_{12} is disjoint either from A or from B.

Without loss of generality assume that C_{12} is disjoint from A. We claim that C_{12} does not intersect the region bounded by W_A. Indeed, because C_{12} is disjoint from A, then it is also disjoint from W_A as $W_A = A \cup Q_{12}(z_i, z_{i+1})$ and C_{12} is disjoint from $Q_{12} \supset Q_{12}(z_i, z_{i+1})$. It follows that if C_{12} intersect the region bounded by W_A, then it is contained in it. But this is impossible because the region bounded by C_{12} contains, by assumption, the curve Q_{12}.

Next, we claim that neither v_3 nor v_4 can be located in the region bounded by W_A. This is because, by Claim A.1, as we traverse Q_{34} from say v_3 in the direction of v_4 we must hit the curve C_{12} before hitting x_3, the first intersection point we encounter with Q_{12}. However, if v_3 is located in the region bounded by W_A, then because C_{12} does not intersect this region it follows that as we traverse Q_{34} from say v_3 in the direction of v_4 we must first hit a point of W_A. Of course we cannot hit a point of A because $A \subset C_{34}$ and Q_{34} is disjoint from C_{34}. Therefore, we must hit a point of $W_A \setminus A = Q_{12}(z_i, z_{i+1}) \subset Q_{12}$. This contradicts our assumption. Similarly, we show that v_4 is not contained in the region bounded by W_A. It now follows from Jordan's theorem that Q_{34} meets W_A an even number of times. This implies that Q_{34} meets $Q_{12}(z_i, z_{i+1})$ an even number of times, as Q_{34} is disjoint from A.

We have thus reached the desired contradiction showing that Q_{34} meets each of the portions $Q_{12}(z_i, z_{i+1})$ an even number of times and therefore Q_{34} meets the entire curve Q_{12} an even number of times, contrary to our assumption.

We note that the proof applies to families of pseudo-line as well as pseudo-circles. A family $\mathcal{C}$ of curves in the plane is called a family of pseudo-lines if every two curves meet at most once and (very importantly for us) the ends of each curve in $\mathcal{C}$ are at infinity. A family of

lines in the plane is an example for such a family.

By applying the inversion mapping that takes z in the complex plane to $\frac{1}{z}$ (assuming that the origin does not belong to any of the curves) a family of pseudo-lines becomes a family of (pairwise intersecting) pseudocircles meeting each other also at the origin. Therefore, the proof for pseudo-lines follows immediately.