Fault Recovery in Wireless Networks: The Geometric Recolouring Approach

Henk Meijer[1], Yurai Núñez-Rodríguez[2], and David Rappaport[2] ⋆

[1] Roosevelt Academy, Middelburg, The Netherlands
h.meijer@roac.nl
[2] Queen's University, Kingston, ON Canada
{yurai,daver}@cs.queensu.ca

Abstract. Duplication of information allows distributed systems to recover from data errors, or faults. If faults occur spontaneously, without notification, and disguised incorrect data blends in with correct data, their detection becomes non-trivial. Known solutions for fault recovery use monitoring mechanisms that compare the data in multiple nodes to infer the occurrence of faults. To this end, we propose a localized geometric approach to fault recovery in wireless networks. We compare our approach with a more traditional combinatorial approach that uses a majority rule. Our experiments show that our geometric approach is an improvement over the majority rule in some cases, whereas in the other cases a hybrid method that combines the best of both strategies is superior to each individual method.

Key words: fault recovery, recolouring, wireless networks, computational geometry, experimental algorithms, localized algorithms

1 Introduction

We consider fault recovery in networks consisting of disambiguating two-state variables at the nodes. This can be achieved by using information duplicates stored across the network. Our results easily generalize to multiple-state variables if applied independently to their constituent bits. We examine a geometric technique that provides autonomous detection and recovery of faults. We compare our technique with an existing non-geometric approach and demonstrate its effectiveness. The geometric method we propose may not be the only effective one; so, with this work we hope to open the discussion and make progress towards the best localized fault recovery strategy.

In what follows we refer to faulty and healthy nodes as red and blue, without defining which colour is assigned to which state. Colouring the nodes will simplify our presentation and will put our work in the same context as previous work. Thus, the change of status of a node (from faulty to healthy, or vice versa) is called a *recolouring*.

The rest of this document is organized as follows. Section 2 reviews previous work on fault recovery techniques for distributed systems and the origins of a

⋆ Supported by an NSERC of Canada Discovery Grant technique we propose for fault recovery, namely, geometric recolouring. Section 3 presents the details of fault recovery using our geometric approach. In Section 4 we present our experimental results and demonstrate the suitability of geometric recolouring for fault recovery. Finally, we identify a set of open problems derived from our work.

2 Previous Work

We first survey previous work on fault recovery in distributed systems. Then we introduce the fundaments of a geometric recolouring technique as used for a related problem, the red-blue separation problem.

Several studies on fault recovery have focused on distributed systems with regular topology. For example, Floccini et al. [2] consider fault recovery on toroidal meshes and Luccio, Pagli, and Sanossian [5] study this problem on butterfly networks. Their work focuses on finding *monopolies*, that is, configurations of faulty nodes that can cause the entire network to fail. As will be seen later on, we are more concerned with the length of recolouring sequences. The underlying goal of both efforts remains to study systems that can autonomously recover from faults through localized algorithms.

It is often assumed that faults occur as a consequence of manufacturing defects or other random, spontaneous causes. This is the case considered by Krishnamachari and Iyengar [4]. To our knowledge, no previous work has considered the occurrence of failures in correlation with the geometric location of the nodes.

It is widely accepted that the data collected by the nodes of a network, in the case of sensor networks for instance, is correlated to their geographic location (see [11, 1] for example); thus, there is no reason not to believe that errors induced on the network by the influence of the environment are also correlated to their spatial distribution. Notice also that considering faulty areas is a generalization to considering isolated errors. The latter can be seen as independent faults on areas that are small enough to contain a single node.

It is assumed that a node does not know whether it is faulty or not just by reading its data. It can however, compare its colour (state) with its neighbours' colours. All the previously cited approaches to fault recovery and other studies (see the survey by Peleg [8]) use a *majority voting* rule. The majority voting rule recolours a node if it has more neighbours of the opposite colour. This strategy can obviously turn healthy nodes to faulty as much as healing faulty ones. However, based on the fact that faulty nodes should be a “non-dominant minority”, we expect that the cooperative effort makes progress towards healing the faulty nodes. On the other hand, we argue that the majority rule is not the best approach if applied to geometric graphs involving areas of faulty nodes. We propose a geometric recolouring approach to this end.

Geometric recolouring, as introduced by Reinbacher et al. [10], has been used for assisting geographic data classification. Given a planar set of red and blue points, their goal is to separate the red from the blue by polygonal curves with small perimeter. A reclassification method is performed as a preprocessing step to correct misclassified points, as the input is assumed to possibly contain errors. This reclassification technique is termed recolouring, which we call *geometric recolouring* for reasons that will become evident in what follows. Their experiments show that the use of recolouring yields separating curves with smaller perimeters.

Reinbacher et al. use a Delaunay triangulation of the point set as the underlying structure for recolouring, specifying a neighbour relation among the points. They then define a point, p, as *surrounded* when there is a contiguous set of oppositely coloured neighbours of p, in the triangulation, that span a radial angle greater than 180° (see Figure 1). Points that are surrounded are iteratively recoloured, in no particular order, until no point remains surrounded. This strategy raises an interesting question concerning the finiteness of recolouring sequences. It turns out that if one chooses the threshold angle to consider a point as surrounded as any value smaller than 180°, there may exist trivial infinite recolouring sequences [10]. The problem is far less trivial for thresholds greater than 180°.

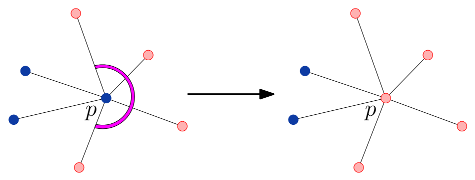

Fig. 1. Recolouring surrounded point p.

Using properties of the triangulation, Reinbacher et al. show that these recolouring sequences are finite and guaranteed to terminate in at most $2^n - 1$ iterations. They also show an example triangulation that yields $O(n^2)$ recolourings. In a previous work [6], we proved that any triangulation can have at most $O(n^2)$ recolourings, closing the gap between the lower and the upper bound. Moreover, we extended our results to other geometric graphs and proved a set of bounds for the length of recolouring sequences, ranging from linear for trees to infinite for planar graphs. In the following section we use one of our polynomial bounds for a convenient construction, the NIC graph.

3 Fault Recovery in Wireless Networks

Before presenting our geometric approach to fault recovery we propose a general framework for recolouring in wireless networks. We review a more traditional combinatorial approach to recolouring. Then we present the challenges of our geometric approach, along with proposed solutions. Finally, we include a hybrid method that combines the combinatorial and geometric approaches.

A node can be considered as surrounded or dominated by neighbours of the opposite colour according to different criteria. For now, we assume that we are provided with a general function SURROUNDED that returns TRUE for a node surrounded by neighbours of the opposite colour, or FALSE otherwise. Different versions of SURROUNDED will be studied in what follows.

As pointed out earlier, a recolouring does not necessarily mean that a faulty node is being fixed; on the contrary, sometimes a healthy node can become faulty by the same mechanism. However, our experiments demonstrate that whenever the cause of the fault affects a relatively small area, the number of nodes that are recovered from faults is substantially larger than the number of nodes that turn faulty. In fact, in many cases all faulty nodes are able to recover (see Section 4 for more details).

The fault recovery algorithm simply consists of the *Recolouring Protocol*, as defined next, to be executed at all nodes. The protocol defines successive recolourings of a node, according to the function SURROUNDED, and a mechanism to notify its neighbours whenever a change of colour occurs.

```
Algorithm 1: Recolouring Algorithm
input  : Network $G = (N, L)$ with bi-chromatic nodes.
output : Network $G = (N, L)$ with a different node colouring such that no more
         nodes can be recoloured.

Recolouring Protocol
Step 1. Broadcast a COLOUR message to all neighbours, with the node's colour
    information.
Step 2. If there is a COLOUR message in the node's queue, the new colour of the
    corresponding neighbour is considered for updating the node's colour according
    to the function SURROUNDED .
Step 2.1. If the node is recoloured, broadcast a message to all neighbours with the
    new colour information.
Step 3. If there is no COLOUR message in the queue and no COLOUR message has
    been received for $T$ time units, go to Step 1.
Step 4. Go to Step 2.
```

Step 3 of the recolouring protocol ensures that if a node becomes faulty, the neighbours are informed of its colour change. The parameter T can be adjusted for an optimal tradeoff between fast response to faults and low network traffic. The algorithm cycles idly (or terminates temporarily) once no more nodes can be recoloured. However, it restarts itself after T time units of inactivity, to recover from possible new faults.

Notice that the nodes do not necessarily wait until they know the colour of all neighbours. The protocol is presented in a way that no synchronization is required. In fact, asynchrony is one key aspect for the termination of the algorithm.

On the other hand, if rounds of recolourings occur synchronously for all surrounded nodes, the process may not terminate. Goles and Olivos [3] explain the behaviour of such systems and how they may fall into configurations that oscillate infinitely.

We have used the concepts of synchrony and rounds in an intuitive, informal way; the reader is referred to Peleg's book [9] for formal definitions. In the following we define other matters of time and synchrony that are relevant to our work. The *recolouring time* of a node p is the time it takes from the actual recolouring, as defined by Step 2 of the algorithm, to the realization by its neighbours that p has been recoloured. We define two recolourings to be *simultaneous* if the corresponding recolouring times overlap. The *sequential model* is defined as a hypothetical model in which no simultaneous recolourings occur, as if there was a global scheduler controlling the network's activity. Last, we define our *asynchronous model* to be one in which the recolourings of nodes occur independently from one another. Thus, simultaneous recolourings happen only as a matter of chance. Next we establish that asynchronous systems behave like the sequential model in the long run with high probability.

Lemma 1. *Let $G = (N, L)$ be a network in which nodes operate asynchronously. The probability that any two nodes $p, q \in N$ recolour simultaneously n times, tends to zero as n tends to infinity.*

Proof. As defined for our synchronous model, the chance that nodes p and q are recoloured simultaneously for the i-th time has the associated probability $P_i(p, q) < 1$. Without considering any particular probability distribution, we can bound $P_i(p, q)$ by $P(p, q)$, the highest probability of simultaneous recolourings of p and q over all colour configurations. Thus, $P_i(p, q) \leq P(p, q) < 1, \forall i$. We can conclude that the joint event consisting of an unbounded number of simultaneous recolourings of p and q has infinitesimal probability, as stated in the following.

$$\lim_{n\to\infty} \prod_{i=1}^{n} P_i(p, q) \leq \lim_{n\to\infty} \prod_{i=1}^{n} P(p, q) = 0$$

because $P(p, q) < 1$, which concludes our proof.

Note that the existence of $P(p, q) < 1$ is guaranteed by the perfect asynchrony assumption. In practical scenarios, if the physical network implementation may cause synchronized behaviour after certain colour configurations, random response times can be introduced to further reinforce the network asynchrony.

The previous lemma proves that long sequences of simultaneous recolourings are unlikely for two or more nodes. Thus, in the long run, a purely asynchronous network behaves (with high probability) like the sequential model. This implies that if the surrounded function of choice ensures finiteness for the recolouring process in the sequential model, it also produces finite recolouring (with high probability) for truly asynchronous environments. Therefore, in the sequel we limit our study of recolouring strategies to the sequential model.

The Surrounded function can be implemented based on simple combinatorial properties of a node and its neighbours. In this case a majority rule is considered. Thus, we define a function Surrounded_combinatorial that returns

TRUE for nodes with more neighbours of the opposite colour than neighbours of its colour and FALSE otherwise. The majority "voting" rule has been applied to a wide variety of dynamic systems, such as cellular automata and other distributed computing systems [2, 4].

In order to prove that this simple strategy terminates, we extend our colouring convention to colour the links of the network. The links connecting either pairs of blue or red nodes are coloured blue or red, respectively. For a connected pair of differently coloured nodes, we mix the colours to obtain a *magenta link*.

Theorem 1. *Let $G = (N, L)$ be a network with bi-chromatic node set N. In the sequential model, the Recolouring Algorithm with parameter T and the* SURROUNDED_COMBINATORIAL function terminates after $O(|L|)$ recolourings from the time of the last fault plus T.

Proof. We use a simple counting argument on the number of magenta links. The number of magenta links is obviously at most $|L|$. After T units of time from the last fault, any notification of colour change (i.e., COLOUR message) is a consequence of a recolouring, as opposed to a delayed broadcast from a node that has become faulty due to environmental factors. Then, with every recolouring, the number of magenta links incident to the recoloured node decreases. Because no other recolouring occur simultaneously, according to our definition of the sequential model, the overall number of magenta links decreases with every recolouring. Thus, at most $O(|L|)$ recolourings can occur, which proves the theorem.

We assume that each node knows all its neighbours and the angle each neighbour is from it. With the angle information at hand, a new technique can be developed for implementing the SURROUNDED function of Algorithm 1. The geometric criterion we use for fault recovery is geometric recolouring as presented in the previous section. That is, the function SURROUNDED_GEOMETRIC is defined to return TRUE if the angle defined by the oppositely coloured neighbours of the node in question is greater than $180°$, and FALSE otherwise. Also, nodes with one neighbour are never considered surrounded, and nodes with all (2 or more) neighbours of the opposite colour are always surrounded, as the surrounding angle is considered $360°$.

In order to guarantee the termination of the geometric recolouring approach to fault recovery, some preprocessing of the network is required. In what follows we provide a set of preliminary results required to introduce the preprocessing algorithm and the geometric approach to fault recovery.

It is known that a geometric recolouring process can be infinite for general (non-planar) networks (see [6]). Thus, the recolouring strategy cannot be directly applied to any network, because it is crucial that the fault recovery process terminates. Instead, it can be applied to a network that approximates the original network as well as possible and guarantees finite recolouring. To this end we use NIC networks, as defined next.

Definition 1. ***(convex node and convex links)*** *A* convex node *p of a network is a node with two consecutive incident links, the* convex links *with respect to p, that define an angle greater than* $180°$.

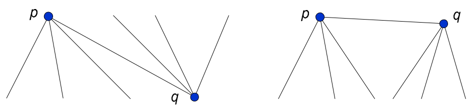

Fig. 2. Two cases of adjacent convex nodes p and q sharing a convex link.

Theorem 2. *Let $G = (N, L)$ be a (not necessarily plane) network with set of bi-chromatic nodes N and set of links L, such that every node p in G satisfies one of the following three conditions:*

- *p has degree less than or equal to 1,*
- *p is not convex,*
- *p is convex and is adjacent to another convex node through a convex link (i.e., p is not an* isolated convex *node).*

The length of geometric recolouring sequences of G is $O(|N||L|)$.

This theorem is a generalization of our bound for geometric recolouring in triangulations from [6]. The full proof can be found in [7].

We define a network that satisfies the conditions of Theorem 2 to be a *NIC network*. One of the advantages of using NIC networks is that they can be described using only local properties of the nodes and therefore, can be computed in a localized manner.

Corollary 3 *Let $G = (N, L)$ be a NIC network with bi-chromatic node set N. In the sequential model, the Recolouring Algorithm with parameter T and the* SURROUNDED_GEOMETRIC *function terminates after $O(|N||L|)$ recolourings from the time of the last fault plus T.*

In what follows, we show how to compute a NIC network that represents, as well as possible, the original structure of the network in a localized manner. The idea is to start with the original network and either "add" or "remove" a small number of links such that the resulting network is a NIC network. By adding and removing links we mean that a node considers new nodes as neighbours or disregards neighbours, only for recolouring purposes.

Through edge additions one could construct a NIC network or even a (non-planar) superset of a triangulation, which is known to have at most a cubic ($O(N^3)$) number of recolourings (Theorem 13 [6]). However, adding links may involve pairs of nodes that are multiple hops away from each other in the network, which calls for non-localized algorithms and more communication intensive distributed computation. The goal then is to remove the minimum number of links so that the network satisfies the NIC conditions. This way the topology of the original network is fairly well preserved. The next theorem states that it is hard to find such optimal configuration, even if centralized computation is allowed. We formally state the problem and the complexity of its decidability version, which directly implies the hardness of its minimization version.

Problem 1. **Non-Isolated Convex (NIC)**
Instance: (G, k), where $G = (N, L)$ is a network with $|N| = n$ and k is a numeric constant.

Question: Is there a set of links $L' \subset L$ such that $|L'| \leq k$ and $G' = (V, L \setminus L')$ satisfies the NIC conditions?

Theorem 4. *The NIC problem (Problem 1) is NP-Complete.*

We omit the proof of this theorem due to limited space. For details see [7].

Because the optimal solution is hard to find, we study heuristic algorithms that eliminate a relatively small number of links. It is noteworthy that in the worst case the optimal number of link removals may be linear in the number of links and quadratic in the number of nodes. An example network that exhibits this complexity consists of a complete bipartite network, where the nodes of each partition lie on one of two parallel lines (see [7] for details).

The example we just described shows that optimal solutions, and heuristic solutions alike, cannot always eliminate a small number of links in the worst case. Therefore, we propose a simple heuristic algorithm that eliminates a small number of links according to our experiments. The heuristic algorithm for constructing the NIC network, the NIC Algorithm (Algorithm 2), is described next. The NIC Algorithm operates under the same assumptions as the geometric recolouring strategy: all nodes know their neighbours and the angle they define with respect to them. The algorithm consists of a single protocol executed at all nodes.

Link marks, as used in the protocol, are relative to the node, that is, a link can be marked differently by each incident node. Notice that Step 3 of the protocol assures that no isolated convex node will remain connected to a non-convex node. The algorithm is guaranteed to terminate because links are always removed and never replaced. Note that a network empty of links satisfies the NIC conditions so, in the worst case the algorithm terminates when all the links have been removed. Obviously, this is a very pessimistic analysis, as normally only a small fraction of the links are removed (see Section 4). Furthermore, this protocol does not fall into infinite loops because once a link is convex, it never becomes non-convex.

We also propose a simple hybrid strategy that combines the combinatorial and the geometric strategies and yields the best experimental results for certain degrees of connectivity, as will be discussed in Section 4. The combination of the combinatorial and geometric recolouring strategies requires some extra care; otherwise the resulting strategy may not terminate, despite each separate strategy does. For *hybrid recolouring* we define a SURROUNDED_HYBRID function that uses the majority rule for the most part, except that when the number of neighbours of the same and opposite colours are equal, the geometric (surrounding angle) criterion is used to break the tie.

It is easy to see that this process yields a finite recolouring sequence if the geometric component considers a NIC subnetwork: the number of magenta links always decreases or remains the same (see the proof of Theorem 1). Also, while

Algorithm 2: NIC Algorithm

```
input  : Network $G = (N, L)$ with bi-chromatic nodes.
output: Network $G' = (N, L')$ such that $L' \subseteq L$ and $G'$ satisfies the NIC
        conditions.

NIC Protocol (executed at node $p$)
Step 1. Mark all links incident to $p$ as unknown.
Step 2. If there is no message in $p$'s message queue and there are still links marked
    as unknown, then send a message to each neighbour. The type of the message
    sent is either CONVEX or NON-CONVEX, depending on the convexity of the
    link with respect to $p$.
Step 3. If there is a message in the node's message queue, process the message
    according to its type:
  CONVEX: the link through which the message was received is marked as convex. If
    the link was not marked as convex before, then a message is sent back to the
    sender indicating the convexity with respect to $p$.
  NON-CONVEX: the link through which the message was received, $l$, is marked as
    non-convex. If $l$ is convex with respect to $p$, and $p$ is an isolated convex node,
    then $p$ removes $l$, sends a REMOVE message to the corresponding neighbour,
    and sends CONVEX messages over any other link that may have become convex
    after removing $l$.
  REMOVE: the link through which the message was received, $l$, is removed.
Step 4. Go to Step 2.
```

recolourings that preserve the number of magenta links occur, geometric recolouring converges for the same reasons as Theorem 2. It then follows that the number of recolourings is at most the multiplication of the maximum possible number of recolourings for each method. This is stated in the following theorem.

Theorem 5. *Let $G = (N, L)$ be a network with bi-chromatic node set N. In the sequential model, the Recolouring Algorithm with parameter T and the* SURROUNDED_HYBRID *function terminates after $O(|N||L|^2)$ recolourings from the time of the last fault plus T.*

4 Experiments

There are aspects of recolouring in networks that make an accurate probabilistic analysis quite complicated; for example, small changes in the recolouring order may produce completely different colour configurations. Thus, we turn to experimentation for our analysis.

We coded our recolouring simulator using Java. The experimental test bed consists of a set of connected networks generated at random with $N = 100$ nodes uniformly distributed over a 100 by 100 square grid with area $A = 10^4$. Two nodes share a link if and only if the distance between them is at most a certain unit Δ, to form what is known as a unit disk graph. We generate a set of 1000 random connected networks with unit distance Δ taking on values 15, 20, 25, and 30 times the width of a grid square. These distances have been chosen so that the network is k-connected with high probability for values of k ranging from 1 to 10: $\Delta = 15$ approximately corresponds to $k = 1$ and $\Delta = 30$ to $k = 10$. The results plotted below are averaged over the 1000 randomly generated networks. Notice that the network size (number of nodes) is not critical for our results, as the phenomena we study affect only localized, relatively small areas of the network. The density of the network, however, does play a crucial role. For this reason, our experiments consider different degrees of connectivity, as explained above.

We first present the experimental results for the NIC algorithm. The mean ratio between number of links remaining and total number of links is plotted in Figure 3 (left) for different transmission radii (Δ). It is noticeable that the results improve as Δ and the network connectivity increase. Obviously, for higher values of Δ the convex nodes tend to appear only at the boundary of the grid. According to the NIC Algorithm these are the only nodes from which incident links are removed. Another measure of interest is the number of messages incurred while computing the NIC network. This result is plotted in Figure 3 (right). The graph shows a super-linear growth in the number of messages with respect to Δ. This is expected, as the connectivity of the network increases, at a nearly quadratic rate with respect to the transmission radius (Δ), as presented by Wang and Yi [12] (Theorem 2). Whether more efficient heuristics can be devised for locally computing NIC networks remains an interesting question.

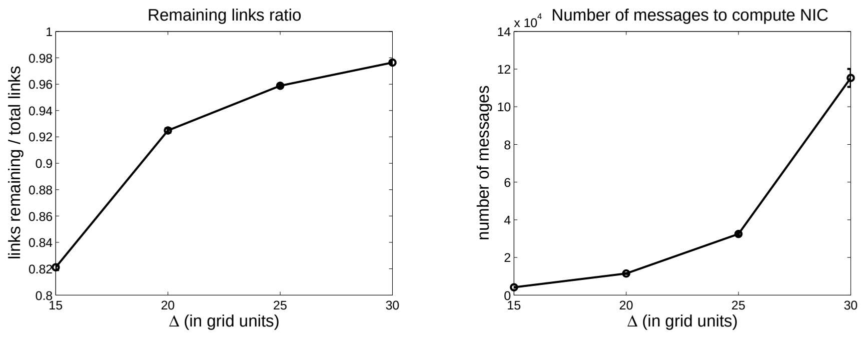

Fig. 3. Results produced by the NIC Algorithm: (left) remaining links ratio, (right) number of messages.

Next we present the results of the recolouring process for all the methods described above. For our experiments we have induced faults on the nodes that fall within a randomly chosen circular area within the grid. The circular area is defined by radii E =10, 12, 14, 16, 18, and 20 times the width of a grid square, and is placed such that it falls at least half its radius away from the border of the grid area. The latter is meant to eliminate the “border effect”, that is, faulty nodes at the border are more difficult to recover because of the smaller number and angle span of neighbours around them. The expected number of faulty nodes is $\pi E^2 N/A$ in our examples.

From our experiments we conclude that for sparse graphs, $\Delta < 25$, the combinatorial and hybrid methods outperform geometric recolouring, with slight advantage to the hybrid method. This comes as no surprise, because on sparse graphs there are not enough links, or angles defined by links, to perform an accurate geometric recolouring. The most interesting results correspond to denser networks, $\Delta > 25$ (see Figure 4 (left) corresponding to $\Delta = 30$). This graph shows the mean ratio of nodes that remain (or become) faulty after the recolouring process terminates.

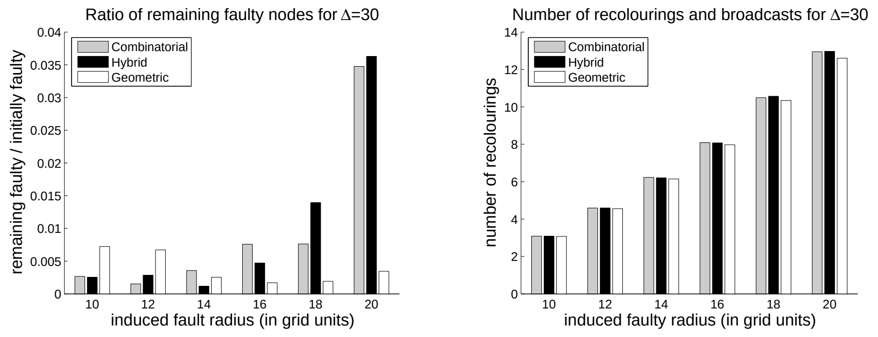

Fig. 4. Results of the recolouring algorithm: (left) fraction of the nodes that remain faulty, (right) number of recolouring and messages incurred.

It is not surprising that geometric recolouring gives better results as the network gets denser and the size of the affected area increases. In such scenarios, the faulty neighbours of a node can sometimes outnumber the healthy ones, which makes the fault spread out of the affected region if using the combinatorial or hybrid methods. On the other hand, as the affected region is convex (a disk in the experiments presented above), no healthy node can be surrounded by faulty ones. This confines the faults to the region initially affected, if not heals the region altogether as it happens in most cases. Figure 4 (right) shows that, despite the smaller number of nodes healed by the combinatorial method, the number of recolourings (and broadcasts) is approximately the same for all methods. This evidences that for large error sizes on dense networks the combinatorial criterion spends many recolourings in turning healthy nodes into faulty, an obviously undesirable effect.

We also conducted experiments where the induced error was defined by a non-convex shape: we used the union of a pair of disks intersecting at a point for our experiments. The results are remarkably similar to the results previously presented for faults induced by single disks. Thus, we conclude that the effectiveness of geometric recolouring is not limited to convexly-shaped affected areas, but also to areas that are partly convex, at the very least.

5 Open Problems

In the previous sections we have mentioned some open problems and conjectures. We summarize a list of these and other problems of interest.

– Compute NIC networks as efficiently as possible, that is, using a small number of messages.
– Characterize networks that have finite geometric recolouring sequences through local properties at the nodes, other than NIC networks.
– Find other recolouring strategies suitable for fault recovery.

References

1. M. Adler, E.D. Demaine, N.J.A. Harvey, and M. Pătraşcu. Lower bounds for asymmetric communication channels and distributed source coding. In *Proc. 17th Annual ACM-SIAM Symp. on Disc. Alg. (SODA'06)*, 251–260, 2006.
2. P. Flocchini, E. Lodi, F. Luccio, L. Pagli, and N. Santoro. Dynamic monopolies in tori. *Disc. Applied Math.*, 137(2):197–212, 2004.
3. E. Goles and J. Olivos. Periodic behaviour of generalized threshold functions. *Disc. Math.*, 30:187–189, 1980.
4. B. Krishnamachari and S. Iyengar. Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Trans. on Comp.*, 53(3):241–250, 2004.
5. F. Luccio, L. Pagli, and H. Sanossian. Irreversible dynamos in butterflies. In *Proc. 6th Intl. Colloq. on Structural Inf. and Comm. Complex.*, 204–218, 1999.
6. H. Meijer, Y. Núñez-Rodríguez, and D. Rappaport. Bounds for point recolouring in geometric graphs. *Comp. Geom.: Theory and Apps.*, 42(6-7):690–703, 2009.
7. Y. Núñez-Rodríguez. Problems on Geometric Graphs with Applications to Wireless Networks. PhD Thesis, School of Computing, Queen's University, 2009. http://qspace.library.queensu.ca/handle/1974/5335
8. D. Peleg. Local majority voting, small coalitions and controlling monopolies in graphs: A review. Technical Report CS96-12, Department Of Mathematics & Computer Science, Weizmann Institute Of Science, 1996.
9. D. Peleg. *Distributed Computing: A Locality-Sensitive Approach.* SIAM Monographs on Disc. Math. and Apps., 2000.
10. I. Reinbacher, M. Benkert, M. van Kreveld, J.S.B. Mitchell, J. Snoeyink, and A. Wolff. Delineating boundaries for imprecise regions. *Algorithmica*, 50(3):386–414, 2008.
11. D. Slepian and J.K. Wolf. Noiseless encodings of correlated information sources. *IEEE Trans. on Inf. Theory*, 19(4):471–480, 1973.
12. P.J. Wan and C.W. Yi. Asymptotic critical transmission radius and critical neighbour number for k-connectivity in wireless ad hoc networks. In *Proc. of the 5th ACM Intl. Symp. on Mobile Ad Hoc Networking and Comp.*, 1–8, 2004.