Design and Analysis of Networks with Large Components in Presence of Region-Based Faults

Sujogya Banerjee, Shahrzad Shirazipourazad, and Arunabha Sen
Computer Science and Engineering Program
School of Computing, Informatics and Decision System Engineering
Arizona State University
Tempe, Arizona 85287
Email: \{sujogya, sshiraz1, asen\}@asu.edu

Abstract—Connectivity \(\kappa(G)\) of a network \(G\) is traditionally considered to be the primary metric for evaluation of its fault-tolerance capability. However, connectivity as a metric has several limitations - e.g., it has no mechanism to distinguish between localized and random faults. Also it does not provide any information about the network state, if the number of failures exceed \(\kappa(G)\). The network state information that might be of interest in such a scenario is the size of the largest connected component. In this paper, we address both these limitations and introduce a new metric called region-based largest component size (RBLCS), that provides the largest size of the component in which the network decomposes once all the nodes of a region fail. We study the computational complexity of finding RBLCS for a given network. In addition, we study the problem of least cost design of a network with a target value of RBLCS. We prove that the optimal design problem is NP-complete and present a heuristic to solve the problem. We evaluate our heuristic by comparing its solutions with the optimal solutions. Experimental results demonstrate that our heuristic produces near optimal solution in a fraction of time needed to find the optimal.

I. INTRODUCTION

The node/link connectivity \(\kappa(G)\) is traditionally considered to be the primary metric for evaluation of the fault-tolerance capability of both wired and wireless networks. However connectivity metric has two major limitations: (i) it only considers node/link failures that are random in nature, i.e, the probability of a node or link failing is independent of its location in the deployment area (this is true because connectivity has no way of capturing the notion of the locality or spatial correlation of the faults), (ii) the connectivity metric fails to provide any information about network state (i.e, the number and size of the components) once the number of failures exceed \(\kappa(G)\).

Most of the traditional studies of fault tolerance in wired and wireless networks [1], [2] assume that the node/link failures are random in nature. However, the assumption of random node/link failure is not valid in many scenarios. This is particularly true in a military environment, where an enemy bomb can destroy a large number of nodes confined in a limited area. This situation is shown in Fig. 1(a) where the shaded part indicates the fault region. The networking research community in recent past has shown considerable interest in studying localized, i.e., spatially correlated or region-based faults in both wired and wireless networks [3]–[9]. The limitation (i) of connectivity metric can be overcome by utilizing the metric region-based connectivity introduced in [5], [8]. A region may be defined either with reference to the network graph (i.e, the topological relationship between the nodes) or with reference to the network geometry (i.e, layout of the nodes and links in a geographical area). There may be many different ways of defining a region with respect to a network graph. For example, a region in a network graph \(G = (V, E)\) may be defined as (i) a subgraph of \(G\) with diameter \(d\) (the maximum of the shortest path distance between a pair of nodes, taken over all source-destination node pairs), or (ii) a subgraph of \(G\), centered in some \(v \in V\) and radius \(r\). With reference to network geometry, a region may be defined as a circular area in the network layout covering a set of nodes and links.

We elaborate limitation (ii) of connectivity metric with an example shown in Fig. 1(b). It can be noted that the connectivity of both the linear array and the star networks is 1. Therefore no distinction between these networks can be made regarding their robustness (or fault-tolerance capability) using connectivity as the metric. However, the following observations can be made regarding the state of these two networks after failure of one node: (i) the linear array network can break up into at most two components and the size of at least one component will be \(\geq \lceil n/2 \rceil\), (ii) the star network can break up into \((n - 1)\) components and the size of these components can be as small as 1, where \(n\) denotes the number of nodes in the network. In the unfortunate event of a network being disconnected after a failure, it is certainly desirable to have a few, large connected components than a large number of small connected components. From operational point of view, a linear array network will certainly be preferable to a star network as it offers the possibility of a graceful performance degradation instead of a catastrophic failure.

This motivates us to introduce a new metric for network fault tolerance called region-based largest component size (RBLCS). It is formally defined as follows:

Definition : Suppose \(\{R_1, \ldots, R_k\}\) is the set of all possible regions of a graph \(G\). Consider a \(k\)-dimensional vector \(C_L\) whose \(i\)-th entry, \(C_L[i]\), indicates the size of the largest connected component in which \(G\) decomposes when all nodes in \(R_i\) fails. Then, region-based largest component size (RBLCS) \(\gamma_R(G)\) of graph \(G\) with region \(R\) is defined as

\[
\gamma_R(G) = \min_{1 \leq i \leq k} C_L[i]
\]

In order to have graceful degradation in performance, we
may want to design networks with a high value of $\gamma_R(G)$. The giant or (largest) connected component of a random graph in fault free environment has been studied in [10], [11]. In this paper we are considering giant components for any arbitrary graph when its layout in a 2-dimensional plane is given and faults are confined to a region. The contributions of the paper are as follows:

- We introduce a new metric to capture network state where the traditional metric connectivity is inadequate.
- We provide a polynomial time algorithm for computing region-based largest component size (RBLCS)
- We consider a network design problem with a target value of RBLCS and show that the problem is NP-complete.
- We provide a heuristic for the design problem.
- We conduct the experimental evaluation of the heuristic and show that it produces near-optimal solutions in a fraction of time needed to find the optimal solution.

A. Ideas behind the RBLCS Algorithm

Before we present the algorithm, we make a few observations on which the algorithm is based. Our algorithm is valid for both wired and wireless networks. In a wired network, a physical link connects two nodes. If a node is destroyed due to failure of a region, all links incident on that node are also destroyed. However, it is possible that failure of a region destroys a link without destroying the nodes at its ends points. In a wireless network, there is no physical link, and as such the possibility of a fault destroying a link does not arise. There could potentially be infinite number of circular regions that covers the 2-dimensional plane where the nodes and links are deployed. It may be noted that a node corresponds to a point in this plane and a line (i.e., a straight line in the plane) and a region (i.e., a circular area in the plane) correspond to a set of points in the plane. We say that region $R$ intersects or covers line $l_i$, if $R \cap l_i \neq \emptyset$. Although there could be an infinite number of circular regions in the plane, for the purpose of computation of RBLCS, we only need to consider a finite number of them. Two regions are said to be indistinguishable if they cover the same set of links and nodes. Otherwise, they are distinguishable or distinct. For computing RBLCS, we only need to evaluate the distinct regions. Since there are $n$ nodes and $m$ links in the network, there could be at most $2^{n+m}$ distinct regions. We will show next that the number of distinct regions that needs to be considered is bounded by a polynomial function of $n$ and $m$. Two indistinguishable regions are shown in Fig. 2.

II. Network Robustness Analysis

In this section, we provide an algorithm that computes the RBLCS of a graph $G = (V, E)$ when its layout in a plane is given as input and the region $R$ is defined to be a circular area with radius $r$. The algorithm computes $\gamma_R(G)$ in $O(n^6)$ time, where $|V| = n$. The inputs to the algorithm are the following:

(i) a graph $G = (V, E)$ where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ are the sets of nodes and links respectively, (ii) the layout of $G$ on a 2-dimensional plane $LG = (P, L)$ where $P = \{p_1, \ldots, p_n\}$ and $L = \{l_1, \ldots, l_m\}$ are the sets of points and straight lines on the 2-dimensional plane (note: (a) there is a one-to-one correspondence between the nodes and points in $V$ and $P$, (b) a one-to-one correspondence between the edges and lines in $E$ and $L$, (c) each $l_i$, $1 \leq i \leq m$ connects two points $p_j$ and $p_k$ in $P$ and does not pass through a third point $p_q$, (iii) a region is defined as a circular area $R$ of radius $r$.

A. Ideas behind the RBLCS Algorithm

Before we present the algorithm, we make a few observations on which the algorithm is based. Our algorithm is valid for both wired and wireless networks. In a wired network, a physical link connects two nodes. If a node is destroyed due to failure of a region, all links incident on that node are also destroyed. However, it is possible that failure of a region destroys a link without destroying the nodes at its ends points. In a wireless network, there is no physical link, and as such the possibility of a fault destroying a link does not arise. There could potentially be infinite number of circular regions that covers the 2-dimensional plane where the nodes and links are deployed. It may be noted that a node corresponds to a point in this plane and a line (i.e., a straight line in the plane) and a region (i.e., a circular area in the plane) correspond to a set of points in the plane. We say that region $R$ intersects or covers line $l_i$, if $R \cap l_i \neq \emptyset$. Although there could be an infinite number of circular regions in the plane, for the purpose of computation of RBLCS, we only need to consider a finite number of them. Two regions are said to be indistinguishable if they cover the same set of links and nodes. Otherwise, they are distinguishable or distinct. For computing RBLCS, we only need to evaluate the distinct regions. Since there are $n$ nodes and $m$ links in the network, there could be at most $2^{n+m}$ distinct regions. We will show next that the number of distinct regions that needs to be considered is bounded by a polynomial function of $n$ and $m$. Two indistinguishable regions are shown in Fig. 2.
regions bounded by the arcs [12], [13]. The cells can be described by a constant number of polynomial inequalities of constant maximum degree $d = 2$ [12], [13]. In Fig. 4(a) a region that covers 2 nodes and 2 links (at least partially) is shown. Fig. 4(b) shows the arrangement of the vulnerability zones of these set of nodes and the links. A cell is highlighted in Fig. 4(b).

**Definition - C-point:** A C-point is an arbitrarily selected point within a cell in $\text{arr}(P)$.

**Definition - Principal Regions:** Any region centered at a C-point will be referred to as a Principal Region (Fig. 4(c)).

Note that Principal Region defined in [5] is a region centered at an I-point. Considering the regions centered only at the I-points will not include all distinct regions. For example consider a network with two nodes $n_1$ and $n_2$. Let the NVZs of these nodes intersect at two points $I_1$ and $I_2$. Regions centered at $I_1$ and $I_2$ will cover both the nodes. But if we consider principal regions to be centered at the C-point corresponding to the three cells made by the NVZs then there will be 3 Principal regions - one covering $n_1$, one covering $n_2$ and other covering both the nodes. So Principal regions considered in this paper is a superset of the Principal regions considered in [5]. Also Principal regions centered at the C-points is the actual set of all possible distinct regions in the network. The previous definition of Principal Region in [5] worked because only limited number of distinct regions needed to be examined for computing region-based connectivity.

**Observation 1:** Given a region $R$, the intersection area of the vulnerability zones of the nodes and links within region $R$ is non-empty [5].

**Observation 2:** If a region $R$ covers a set of nodes and links and $R$ is not centered at one of the C-points, there must be at least one other region centered at one of the C-points that covers all the nodes and links covered by $R$. Accordingly this region will be indistinguishable from $R$. The proof follows the proof of Observation 2 in [5].

**Observation 3:** For computing the RBCDN/RBSCS/RBCLS of the network graph $G$ where the layout $LG$ of $G$ is given as input, only a limited number of distinct regions, i.e., only the Principal Regions need to be examined [5].

**Observation 4:** The maximum number of Principal Regions is $O(n^4)$.

**Proof:** By definition, a Principal Region is a region centered at a C-point and number of C-points is equal to the number of cells in the arrangement $\text{arr}(P)$. As the maximum degree of the set of polynomials $\mathcal{P} = \{P_1, \ldots, P_s\}$ defined over $\mathbb{R}^2$ is $2$ [12], [13], the number of cells in $\text{arr}(P)$ is $O(s^2)$. Since $s$ is $O(m)$, there can be at most $O(m^2)$ or $O(n^4)$ Principal Regions.

**Observation 5:** All the C-points can be computed in $O(n^8)$ time.

**Proof:** Using the results presented in [12], [13], we can compute a set of points $\mathcal{C}$ such that each cell in $\text{arr}(\mathcal{P})$ contains at least one point from $\mathcal{C}$, in time $O(s^3)$ (or $O(m^3)$). As a consequence the overall time-complexity to compute all the C-points is $O(m^3)$ (or $O(n^6)$).

**B. The RBLCS Algorithm**

From the observations above, it is clear that we need to examine only the Principal Regions (i.e., the regions whose centers are at C-points) to compute the RBLCS of a network $G$ (with a layout $LG$ on a 2-dimensional plane) and a circular region $R$ of radius $r$. Since there are only $O(n^4)$ of such regions, we can develop a polynomial time algorithm to compute RBLCS.

**Algorithm 1** Algorithm for Computing RBCLS ($\mathcal{R}_R(G)$) of a network graph $G = (V, E)$ with region $R$

1. **INPUT:** $G = (V, E)$, Graph layout $LG = (F, L)$ and $r$
2. **OUTPUT:** $\mathcal{R}_R(G)$
3. Find the set of C-points using the algorithm sketched in Observation 5
4. For each C-point $r_j$, find $G'_j = (V'_j, E'_j)$ a subgraph of $G$ formed by removing the nodes and edges covered by region $R_j$ centered at $r_j$
5. For each such graph $G'_j = (V'_j, E'_j)$ find the largest connected component $LC_j$ using depth-first search [14]. Let $LC\mathcal{S}_j$ be the size of $LC_j$. For graph $G'_j = (V'_j, E'_j)$
6. $\mathcal{R}_R(G) = \min_j LC\mathcal{S}_j$

**Theorem 1:** The complexity of the Algorithm for computing RBCLS is $O(n^6)$. 

![Fig. 4. A Region, intersection of vulnerability zones and a Principal Region](image-url)
Proof: As mentioned in Observation 5, time complexity of finding all the C-points is of $O(n^6)$, where $n$ is number of nodes. So, time complexity of Step 3 is $O(n^6)$. In Step 2, we have to check all edges in $E$ and all nodes in $V$ if they have intersections with the fault region $R$. So, complexity of Step 4 is $O(n^2)$. Step 5 uses depth-first search to compute the number of connected components and has complexity of $O(|V| + |E|) = O(n^2)$. Step 5 is repeated $O(n^4)$ times. Therefore, the complexity of the Algorithm is $O(n^6)$.

III. ROBUST NETWORK DESIGN

In the previous section we presented a polynomial time algorithm to compute the RBLCS of a graph $G = (V, E)$, when its layout $LG = (P, L)$ on a 2-dimensional plane and the radius $r$ of a circular region $R$ is provided as input. In this section, we study a complementary problem, where the goal is to have the least cost augmentation of an existing network, so that it attains a specific target value of RBLCS. Formal description of the decision version of this problem is given below.

RBLCS Augmentation Problem (RBLCS-AP)

INSTANCE: Given (i) a graph $G = (V, E)$ where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ are the sets of nodes and links respectively, (ii) the layout of $G$ on a two dimensional plane $LG = (P, L)$ where $P = \{p_1, \ldots, p_n\}$ and $L = \{l_1, \ldots, l_m\}$ are the sets of points and lines on the 2-dimensional plane, (iii) region $R$ defined to be a circular area of radius $r$, (iv) cost function $c(e) \in \mathbb{Z}^+$, $\forall e \in E$, where $E$ is complement of the link set $\bar{E}$ (i.e., $\bar{E}$ is comprised of the links not present in $E$, but can be added to the graph $G$), (v) integers $C$ and $K$ ($K \leq n$).

QUESTION: Is it possible to increase the RBLCS of $G$ by $K$ by adding edges to $G$ (from the set $\bar{E}$) so that the total cost of the added links is at most $C$?

A. NP-Completeness Proof of RBLCS-AP

We prove that RBLCS-AP is NP-complete by a transformation from the Hamiltonian Cycle in Planar Graph Problem (HCPGP) which is known to be NP-complete [15]. A Hamiltonian Cycle in an undirected graph $G = (V, E)$ is a simple cycle that includes all the nodes. A graph is a planar if it can be embedded in a plane by mapping each node to a unique point in the plane and each edge is a line connecting its endpoints, so that no two lines meet except at a common endpoint [15].

Hamiltonian Cycle in Planar Graph Problem (HCPGP)

INSTANCE: Given an undirected planar graph $G = (V, E)$.

QUESTION: Does $G$ contain a Hamiltonian Cycle?

Theorem 2: RBLCS-AP is NP-complete.

Proof: It is easy to verify whether a set of additional edges of total cost $\leq C$ increases the RBLCS of graph $G$ with region $R$ from $\gamma_R(G)$ to $\gamma_R(G) + K$. Therefore RBLCS-AP is in NP.

From an instance of the HCPGP (a planar graph $G = (V, E)$) we create an instance of the RBLCS-AP (the layout $LG' = (P', L')$ of a graph $G' = (V', E')$) in the following way. First, we do a straight line embedding of the planar graph $G$ on a plane so that lines corresponding to links in $G$ do not intersect each other. Such an embedding can be carried out in polynomial time [16]. We call this layout $LG'' = (P'', L'')$. We create the layout $LG' = (P', L')$ of $G'$ with region $R$ 1. We set the parameters $C$ and $K$ of the instance of the RBLCS-AP to be equal to $n$ and $n - 2$ respectively. We assign costs to the links of $E'$ in the following way. The cost of a link $c(e) = 1$, if $e \in (E' \cap \bar{E'})$ and $c(e) = \infty$, if $e \in (\bar{E} \cap E')$.

If the instance of the HCPGP has a Hamiltonian Cycle, we can use the set of links that make up the cycle, to augment the link set $E'$ of the instance $G' = (V', E')$ of the RBLCS-AP. The augmented $G'$, $(G'_{aug})$, is now a simple cycle that involves all the nodes. With the given definition of region $R$ (a small circle of radius $r$), only one node can be destroyed when a region fails. Accordingly RBLCS of $G'_{aug}$ is $n - 1$. It may be recalled that RBLCS of $G'$ is $1$. Accordingly, augmentation of the link set of $G'$ augmented its RBLCS by $n - 2$. Due to the specific cost assignment rule of the links, the total cost of link augmentation is $n$. Therefore, if the HCPGP instance has a Hamiltonian Cycle, the RBLCS of the instance of RBLCS-AP can be augmented by $K$ with augmentation cost $\leq C$.

Suppose that it is possible to augment the RBLCS of the instance of RBLCS-AP by $K$ with augmentation cost being at most $C$. This implies that the RBLCS of $G'$ can be increased from 1 to $n - 1$ (as $K = n - 2$) when it is augmented with additional links with total cost at most $n$ (as $C = n$). In order for the RBLCS of $G'_{aug}$ to be $n - 1$, the node connectivity of $G'_{aug}$ must be at least 2. A $n$ node graph that has the fewest number of links and yet is 2-connected, is a cycle that includes all the nodes. As $G'$ had no links, this implies at least $n$ links must have been added to create the augmented graph $G'_{aug}$. Given that the cost of a link $c(e) = 1$, if $e \in (\bar{E} \cap \bar{E'})$ and $c(e) = \infty$, if $e \in (\bar{E} \cap E')$, and total cost of link augmentation is at most $n$, it is clear that the links used in augmenting $G$
must be from the set \((E \cap E')\). These links are part of the edge set of the instance of \(HCPGP\). Accordingly, the instance of \(HCPGP\) must have a Hamiltonian Cycle.

**IV. A HEURISTIC FOR RBLCS-AP**

In this section, we propose a heuristic for RBLCS-AP. The notations used in the heuristic are as follows:

- \(G_i\): The subgraph induced from \(G = (V, E)\) by removing links and the nodes covered by the region \(R_i\).
- \(\mathcal{PE}\): The potential link set \(\mathcal{PE} \subseteq E\) is the set of links, such that any link in \(\mathcal{PE}\) can be used to connect two components of \(G_i\).
- \(\mathcal{C}_i\): The set of connected components of \(G_i\). For each connected component \(p \in \mathcal{C}_i\) let \(size(p)\) be the number of nodes in the component, \(AE(p)\) be the set of potential links used to make the component \(p\) and \(cost(p)\) be a cost assigned to the component depending on the set \(AE(p)\). Initially, \(AE(p)\) is empty and \(cost(p)\) is zero.
- \(LCS(i)\): The largest component of the \(G_i\).
- \(\mathcal{PE}'\): The largest component size of the \(G_i\).
- \(B\): The desired largest component size for all regions, i.e., \(B = \gamma R(G) + K\).
- \(l\): The number of regions \(R_i\) for which the graph \(G_i\) has \(LCS(i) < B\).
- \(\mathcal{PE} = \bigcup_{i=1}^{l} \mathcal{PE}_i\).
- \(\mathcal{PE}'\): \(\mathcal{PE}' \subseteq \mathcal{PE}\) is the output of the Heuristic such that \(\sum_{e \in \mathcal{PE}'} c(e)\) is minimum and \(G_i\) has at least one component of size \(\geq B\), \(\forall 1 \leq i \leq l\).
- \(\varphi(e)\): The cost per unit size of the new components (across all regions) that can be created by adding \(e\).
- \(cp_i(e)\): The composite component resulted by adding the edge \(e\) connecting two components \(p\) and \(q\) of \(G_i\). Then, \(cost(cp_i(e)) = (size(p) + size(q)) \times \varphi(e)\) and \(AE(cp_i(e)) = AE(p) \cup AE(q) \cup \{e\}\).

The Heuristic runs through a number of iterations. In each iteration, an edge \(e \in \mathcal{PE}\) is selected in the following way. First for each edge \(e \in \mathcal{PE}, \varphi(e)\) is computed:

\[
\varphi(e) = \frac{c(e) + \sum_{i \in \mathcal{PE}_i} \{cost(p_i) + cost(q_i)\}}{\sum_{i \in \mathcal{PE}_i} \{size(p_i) + size(q_i)\}},
\]

where \(p_i\) and \(q_i\) are the two components of \(G_i\) that can be connected by the edge \(e\). In Algorithm 3 we describe how \(\varphi(e)\) can be calculated for each edge \(e\). Then an edge \(e_b\) with minimum cost per unit size is selected. For all regions, where \(e_b \in \mathcal{PE}_i\), the corresponding set \(\mathcal{C}_i\) is updated. If \(e_b\) was connecting two components \(p\) and \(q\) in \(\mathcal{C}_i\), then \(\mathcal{C}_i\) is updated by replacing \(p\) and \(q\) with the combined component \(cp_i(e_b)\). For each newly created component \(cp_i(e_b)\) in the current iteration we set \(cost(cp_i(e_b)) = size(cp_i(e_b)) \times \varphi(e_b)\). If the updated \(LCS_i \geq B\) then the edges in this new composite component is added to \(\mathcal{PE}'\) and the potential edges of this region is not considered in subsequent iterations. Before continuing to the next iteration the edges in every remaining \(\mathcal{PE}_i\), connecting the same pair of components as \(e_b\), are removed. Since after adding \(e_b\) these edges connect the nodes within one component and cannot increase the size of the components of \(G_i\) anymore.

The algorithm continues until all regions has a component of size larger than \(B\).

It should be noted that even if after adding all the edges in \(\mathcal{PE}\) to the graph \(G\) there exists some regions that do not have a component with size \(\geq B\) then it is infeasible to make RBLCS \(\geq B\). In the worst case the heuristic will add all the edges in \(\mathcal{PE}\). So it will always find a solution to the RBLCS-AP if a feasible solution exists.

**Theorem 3:** The time complexity of the Heuristic is \(O(n^8)\).

**Proof:** In order to find \(\varphi(e)\) for an edge \(e\) we have to consider all the \(l\) distinct regions which is \(O(n^4)\). So, finding the edge with minimum \(\varphi\) (line 5) is \(O(|\mathcal{PE}| \times n^4) = O(n^8)\). In RBLCS-AP \(|\mathcal{E}| = O(n^2)\), the maximum number of links that can be added to the graph \(G = (V, E)\). The for-loop in line 6 is repeated \(l\) times. Also, the while-loop in line 4 is repeated at
Fig. 6. Comparison of the solution of the Heuristic with the Optimal by taking its ratio to the optimal.

most $|\mathcal{PE}|$ times. Therefore, the complexity of the Heuristic will be $O(n^2(n^6 + n^4)) = O(n^8)$.

V. EXPERIMENTAL RESULTS

In this section we present the experimental results of the Heuristic proposed in Section IV. We compare the results of the heuristic with the optimal solution through extensive simulations. We find the optimal solution through exhaustive search method. We generate several random instances of network layout in 2-dimensional plane. The number of nodes, $n$, in these instances are varied from 10 to 30 in a step of 5. For each value of $n$ we create 10 instances where in every instance the node locations are uniformly and randomly distributed on a square deployment area of side length 100 units. The edge between every two nodes is added with probability $p = 0.5$. The heuristic finds a solution within a wide range of choices of $p$ and $n$. However, when $p < 0.5$ and $n > 30$ the optimal algorithm takes an unacceptably long time (several days) to complete. This is due to the fact that when $p < 0.5$ and $n > 30$ the set of the potential edges becomes large. Since the complexity of the optimal algorithm is exponential, with a large set of potential edges, it takes an unacceptably long computation time. Since our goal was to compare performance of the heuristic with the optimal, we restricted our experiments to $p = 0.5$ and $n < 30$.

For every instance, we find the set of R-points, $G'_i$, $C_i$ and $\mathcal{PE}_i$ by using the Algorithm for Computing RBLCS presented in Section II. For each instance we execute the algorithm for values of $K$ (number by which RBLCS needs to be augmented) equal to 1 and 2. For $n > 20$ and $p = 0.5$, almost all the instances do not have a feasible solution for values of $K \geq 3$. Therefore we restrict the value of $K$ to be 1 and 2 only. The results of the experiments are shown in Fig. 6. We plot the average of the ratios of the cost of the solution of the heuristic to the optimal cost for each value of $n$ and $K$. It can be observed that in all of these cases, the ratios are less than 1.4, indicating the superior quality of the heuristic solution. The heuristic executes in a small fraction of time needed to compute the optimal solution in all our experiments.

VI. CONCLUSION

In this paper we introduced, region-based largest component size, a new metric of fault tolerance of a network. We presented an algorithm for computing RBLCS for a given network. We study the problem of designing a network with a target value of RBLCS and present a heuristic for its solution. Future work of this study includes possible reduction of time-complexity $O(n^8)$ of the heuristic so that the heuristic can be applied on very large networks. Also the work can be extended by studying the use of other metrics, like region-based component decomposition number and region-based smallest component size, in measuring the robustness of a network in presence of region-based fault.

REFERENCES