1 Introduction

Increasingly researchers in several domains are realizing the importance of understanding a class of networks as they arise in the physical world, in the virtual world and in society. This is giving rise to a phenomenological study of networks often termed as the new science of networks or complex networks. A common characteristic shared amongst complex networks in several domains is that they are self-organizing, which leads to interesting properties about their structure and organization. Two structural features, which have made them popular, are that they are scale-free and exhibit the small-world phenomena. These features do not occur in simple networks such as lattices or random graphs.

A fundamental question then to address is what governs this self-organization, i.e., why do we find similar structures in diverse applications. Is there a common, underlying governing principle that determines this organization? [6] addresses several of these questions. In particular it is proposed that a complex system optimizes its network structure (or topology) in order to maximize an overall survival fitness function. This function is defined in terms of conflicting measures of efficiency, robustness and cost, and an environmental selection pressure. The primary result of the work in [6] is that fitness maximization by adaptation leads to the spontaneous emergence of optimal network structures, both power law and non-power law, of various topologies depending on the selection pressure. The result is obtained by defining robustness and efficiency in terms of graph theoretical notions and uses genetic algorithms to find a maxima for the fitness function. Their results show that when efficiency is paramount the emergent topology is that of a "Star" and when robustness is important the emergent topology is that of a "Circle". In [5], results are further generalized to include circular skip lists in case of robustness. In [4] such optimal topologies are considered for a distributed hash table with robustness defined in terms of load balancing of requests among nodes, efficiency and cost defined accordingly.

In [6], genetic algorithms are employed during the design phase to find an optimal topology which has maximum efficiency and robustness. However, reconfiguration changes to topology as new input becomes available are not taken into account. In a dynamic complex network, for instance a supply chain network or a communication network, the current topology may need to be reconfigured/optimized given a sequence of requests to transfer goods from source A to source B. Further, such reconfiguration may need to be done in an online fashion, i.e., as the input sequence becomes available. In such scenarios it may not be easy to predict the sequence of events in advance. In Section 2 we describe a framework for which allows us to reconfigure an optimal topology as new demand on the network becomes available. We also consider changes in demand which require additions and deletions of nodes. In Section3 we consider efficiency issues related to complex networks. We consider this issue in the context of human-engineered complex systems in which rapid topology changes occur. It is useful in such scenarios to quickly determine robustness and efficiency of a given complex network. Since human-engineered complex networks can rapidly grow to millions of nodes it is often impractical to quickly measure robustness and efficiency as the network evolves because of high computational complexity of the algorithms. In Section 3, we consider sampling techniques which help us quickly estimate robustness and efficiency within an ϵ bound.

In 1 we discover networks that are either robust or efficient or inexpensive based on weights assigned to each of these factors. However, often it is important to stay in state from which we can reconfigure to one of several desired (most robust, most efficient or most inexpensive) configurations with low transition costs. We term such networks as adaptive networks which allow us to move to configuration with a minimum guaranteed efficiency, or guaranteed robustness with low transition costs. We discuss issues related to adaptive networks in Section 4.

2 Reconfiguration of a Complex Networks

Problem: Reconfiguration of a complex network (cn) may be required when there is a significant change in the workload. The reconfiguration can be done in two ways

- Given the current topology, reconfigure it so that it achieves lower efficiency and robustness under the new workload.
- Given the current topology, add/delete nodes to achieve a new optimal topology.

The workload itself might be known entirely in advance or might only be available one event at a time. Further it might be predictable or unpredictable. In the following subsection, we provide an abstract model based on metrical task systems [1] that captures the problems defined above and in the next one discuss issues in applying this model.

2.1 Model

Given a data model, let $D = \{o_1, \ldots, o_n\}$ be the set of all possible topologies that can be constructed given a set of vertices $V = \{v_1, \ldots, v_n\}$. A graph instance is a layout of nodes and edges subject to a cost constraint T and is termed as a *configuration*. Let $\mathcal{S} = \{S_1, \ldots, S_N\}$ be the set of all possible configurations on D. The cost of processing a request q in a configuration S_i is denoted $q(S_i)$ (if q cannot be processed in S_i we set $q(S_i) = \infty$). Often it is necessary to change configurations to reduce costs for processing requests. The cost for *transitioning* between any two given configurations is given by the function $d : \mathcal{S} \times \mathcal{S} \rightarrow \Re^+$. d is any function that satisfies the following properties:

1. $d(S_i, S_j) \geq 0, \forall i \neq j, S_i, S_j \in \mathcal{S}$ (positivity);
2. $d(S_i, S_i) = 0, \forall i \in \mathcal{S}$ (reflexivity); and
3. $d(S_i, S_j) + d(S_j, S_k) \geq d(S_i, S_k), \forall S_i, S_j, S_k \in \mathcal{S}$ (triangle inequality)

In particular, d does not satisfy the symmetry property, *i.e.,* $\exists S_i, S_j \in \mathcal{S}\ d(S_i, S_j) \neq d(S_j, S_i)$. This asymmetry in transition costs exists because the sequence of operations (i.e. insertion or deletion) required for making configuration changes in a complex network exhibit different costs. Given $\sigma = q_1, \ldots, q_n$, a finite sequence of requests, the objective is to obtain a sequence of configurations $S = (S_0, S_1, ..., S_n), S_i \in \mathcal{S}$ such that the total cost of σ under S is minimized. The total cost is defined as:

$$cost(\sigma, S) = \sum_{i=1}^{n} q_i(S_i) + \sum_{i=0}^{n-1} d(S_i, S_{i+1}), \tag{1}$$

in which the first term is the sum of costs of each request in σ under the corresponding configuration and the second term is the total cost to transition between configurations in S. Note, if $S_{i+1} = S_i$ there is no real change in the configuration schedule and incurred transition cost is zero. An offline optimal algorithm OPT knows the entire σ and obtains a configuration schedule S with the minimum cost. An online algorithm ALG determines $S = (S_0, ..., S_n)$ without seeing the complete request sequence $\sigma = (q_1, ..., q_n)$. Thus ALG determines each physical configuration S_i, based on the so far known workload $(q_1, ..., q_i)$ and makes no assumptions about the future workload in advance (unpredictable). We are not considering the case when the workload can be predicted in the future but that can also be a problem.

2.2 Issues

There are several issues in applying this abstract model to a real-world scenario. Some of the immediate questions to answer are: a) Does d satisfy all the properties in the application domain? If not, what is the nature of d? b) Defining robustness and efficiency for a real application. c) Developing a workload that simulates demand requests and their arrival sequence. d) Do we know all the topology configurations in advance? Do we need to reuse GA to discover those? How? e) Not all topology configurations are feasible even if we enumerate them. For instance in a supply chain once we construct a warehouse we cannot return to a configuration which does not use the warehouse. Thus in these cases we cannot return to a previous configuration. f) Do we apriori know transition costs and query execution costs or do we estimate them? All these questions depend upon the chosen domain.

3 Approximating Robustness and Efficiency in Evolving Complex Networks

Problem: In [6], given a graph $G(V, E)$, the distance $d(i, j)$ between vertices i and j of a graph is defined as the length of the shortest path between the two vertices. The average path length of a graph G is then the average distance between any pair of vertices defined as:

$$\langle d \rangle = AverageAPSP = \frac{\sum_{i,j} d(i,j)}{n(n-1)}, 1 < i, j, < n \tag{2}$$

. Efficiency is then defined as inverse of average vertex-vertex distance:

$$Eff = \frac{1}{\langle d \rangle} \tag{3}$$

In summary, to compute efficiency one needs to compute APSP over the entire graph. The APSP problem can be solved by various algorithms in time $\mathcal{O}(nm+n^2logn)$, $\mathcal{O}(n^3)$, or more quickly using fast matrix multiplication techniques. The APSP problem can be solved in average case in time $\mathcal{O}(n^2logn)$ for various types of random graphs [2].

Robustness: In [6] effective accessibility of a graph is defined as the sum of the accessibilities of its strongly connected components. Structural robustness with respect to vertex j of a graph as the ratio of the effective accessibility of the graph S_j obtained by deleting vertex j from the original graph to the maximum possible effective accessibility for S_j. The average-case structural robustness of a graph is defined as the average of the structural robustness values computed over all the vertices. Please see [6] for exact definitions.

Tarjan's algorithm [7] for finding the strongly connected components of a graph does a depth-first search beginning from a start node. The strongly connected components form the subtrees of the search tree, the roots of which are the roots of the strongly connected components. The complexity is same as of DFS: $\mathcal{O}(|V| + |E|$. Robustness thus has complexity $\mathcal{O}V(|V| + |E|)$. Because these results are slow, specialized, or (with fast matrix multiplication) complicated and impractical, and because recent applications of social network theory to the internet may involve graphs with millions of vertices, it is of interest to consider faster approximations.

There are two general ways to approximate robustness and efficiency. One way is to randomly select nodes and perform APSP or DFS from these nodes. The other alternative is to construct a smaller sample of the graph by explore the structure of the graph through random walks or breadth first search and do APSP or DFS on this reduced graph. While these measures might be sufficient to guarantee average efficiency and average robustness, they might not be sufficient to guarantee maximum robustness and maximum efficiency.

Using these methods can we provide a bound on the max/min measures? Approaches: We are aware of three methods that might help us fastly approximate robustness and efficiency. These include random node selection, random jump, and forest fire. We summarize these methods below: **Random Node:** Uniformly at random select a set of nodes N and a sample is then a graph induced by the set of nodes N.

Random Jump: Uniformly at random pick a starting node and then simulate a random walk on the graph. At every step with probability $c = 0.15$ (the value commonly used in literature) fly back to any node and re-start the random walk. The random walk is done for $100 * n$ steps.

Forest Fire: Randomly pick a seed node and begin "burning" outgoing links and the corresponding nodes. If a link gets burned, the node at the other endpoint gets a chance to burn its own links, and so on recursively. Model has two parameters: forward (pf) and backward (pb) burning probability. The exact definition is given in [3]

We have conducted experiments that compare these three methods for robustness and efficiency. Issues: An immediate issue is describing what robustness and efficiency imply in digital networks especially social networks and how can approximate measures help us. Can we get real data for these vast digital networks?

4 Adaptive Complex Networks

[[This section is copied from emails that Amitabh sent]] We would like to extend the evolutionary approach to discover networks that are not only efficient, robust, inexpensive, but also adaptive. By an adaptive network we imply one that we can reconfigure to one of several desired configurations with low transition costs. For instance, one desired configuration may want a minimum guaranteed efficiency, and another may want guaranteed robustness.

We believe that such adaptive networks will be useful in applications like network centric warfare, adaptive supply chain systems, labor supply networks in information technology, disaster management, simulating terrorist cell networks, etc.

We now want to have two measures of cost: infrastructure cost and transition cost. The first is the number of edges in the network. The second in the number of edges that need to be reconfigured to move to a desired configuration. In general, all edges in the network may not be explicitly required for a given desired configuration. In other words, adaptive networks may have higher infrastructure costs than each of the desired configurations.

Three problems can be formulated in this framework.

Problem 1: We know that for infrastructure cost of n (or n-1), a circle has maximum robustness, and a star has maximum efficiency. What is the best adaptive network that has an infrastructure cost of n, and the minimum transition cost of moving to either a circle or a star?

I think it is the star with two-edge spokes (see Figure 1). To move to a regular star about n/2 edges have to be re-arranged, and the same number to move to a circle. Can we come up with an evolutionary mechanism that discovers this network?

Problem 2: The same as Problem 1, but we allow higher infrastructure costs. For instance, if the infrastructure cost is (n+n/3) we can get a star in which each spoke is actually a cycle or circle of 4 nodes. There are n/3 such spoke-circles. All such spoke-circles share one common node—the center of the star. To transition to a regular star this takes n/3 edge re-configurations, and similarly to transition to a regular circle. Can we discover this configuration.

Problem 3: Generalizes Problem 2. Given an infrastructure cost C(n), an efficiency requirement E(n), and an robustness requirement R(n), discover a network with infrastructure cost at most C(n) and which takes equal cost in moving to (1) a network with efficiency E(n) and (2) and network with robustness R(n).

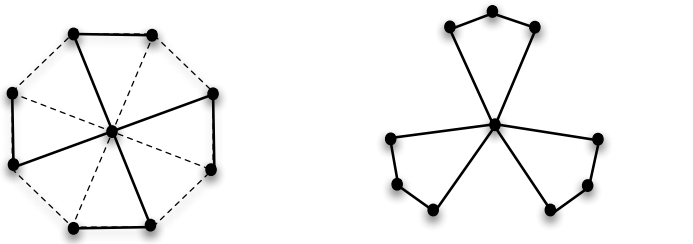

Figure 1: Star with two-edge spokes

4.1 Issues

Several questions are of interest.

1. What are useful definitions of adaptability?
2. How do we handle possible service disruptions during transition? Are notions of overshooting from control theory helpful here?
3. Develop efficient genetic algorithms for discovering adapative solutions.

5 Future Work/Further Questions

: Here we provide a discussion of future work that can be done in this area. It is in response to a RFP.. These questions were raised by Amitabh after a discussion with Venkat. Two aspects are most interesting: A) How can we characterize and compare networks based on their behavior in the case of an attack. B) How can we characterize the changes in a network as a function of the time elapsed after attack. For instance, in a communication network, the traffic patterns after an attack are unprecedented, causing further stress. Or in a supply network, the demand earlier catered to by nodes which have been attacked can migrate to other nodes, possibly overloading them.

In A, it would be interesting to design measures, such as efficiency or robustness, but those which are relevant to a multi-node attack and the resultant changing user response. These will be richer when we also consider B.

One way to model the effects of an attack on a supply network would be to consider a layer of a SUPPLY NETWORK, and a second layer of a DEMAND NETWORK. The supply network is the regular network through which supplies flow: factories, road/rail, warehouses, retail outlets, etc. The demand network is the one through which demands are expected to move if they are not satisfied where they originate. The demand for gasoline, e.g., migrates through the road network. The demand for H1N1 vaccine may migrate through the airline network in addition to the road network. The demand for electricity, on the other hand, cannot migrate. The demand for cell phone connectivity also will not migrate geographically, it will transform into demand for internet connectivity or wired phone connectivity.

So we could propose the following:

1. Identifying network properties which are desirable for different stages after an attack: pre-attack, attack, identifying extent of damage and regrouping, reconfiguration and recovery, rebuilding, etc. Some of the abstract properties we will attempt to formally define are the following.

a) Efficiency b) Connectedness on attack (in this and other such properties we are looking at adversarial attacks rather than random attacks—which should distinguish it from other applications in engineering.) c) Efficiency on attack. d) Relationship between demand and supply networks e) Above relationship on attack—as a function of time elapsed since attack. f) Ease of reconfiguration g) Ease of central control h) Ease of distributed control i) Ease of hierarchical control j) Ease of recovery

2. Identifying networks which have a specified subset of the desirable properties. Here our past work in using evolutionary algorithms will be useful. Further identifying networks which may not have all desirable properties but can be reconfigured easily to move from one to the other.

3. Initiating the study of coupled networks: like demand and supply networks, social-technological networks, or IT labor demand and supply. What are their properties? How can we study their relationships and cross-over properties? How can they be used to model WMD attacks?

4. Studying network properties as a function of time when the network is stressed.

Some other possible directions:

5. Responding to an attack: Prune parts of the network which may not have been attacked, but because we cannot hope to satisfy the associated demand.

6. Related to above: identifying which part of the network should be repaired first during recovery.

7. Developing a recovery strategy which is robust to a second or third attack.

References Cited

[1] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press New York, NY, USA, 1998.

[2] D. Eppstein and J. Wang. Fast approximation of centrality. *Journal of Graph Algorithms and Applications*, 8(1):39–45, 2004.

[3] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, page 187. ACM, 2005.

[4] S. Patil, S. Srinivasa, S. Mukherjee, A.R. Rachakonda, and V. Venkatasubramanian. Breeding Diameter-Optimal Topologies for Distributed Indexes. *Complex Systems*, 18(2):01.

[5] Venkat Venkatasubramanian Sanket Patil, Srinath Srinivasa. Classes of optimal network topologies under multiple efficiency and robustness constraints. 2009.

[6] Venkat Venkatasubramanian, Santhoji Katare, Priyan R. Patkar, and Fangping Mu. Spontaneous emergence of complex optimal networks through evolutionary adaptation, 2004.

[7] Wikipedia.