Resource Allocations on Networks

By Brian Karrer

SI 708

Introduction:

My project deals with the problem of distributing resources over a network. The nodes in the network represent population centers that desire access to a resource. The edges represent possible distribution routes of the resource over the network. In particular, we will focus on disaster management resources, albeit in an abstract mathematical format.

Resource allocation problems fall under the field of Operations Research (OR). Operations researchers make mathematical models designed to help policymakers. The standard resource allocation problem is to distribute a scarce resource across several activities or production units. In our case, we will be distributing disaster management resources across nodes in a network. The standard allocation model, with continuous variables, is solved through linear programming (ref 1). It turns out through some manipulation we will be able to turn our problem into a form very close to that of the OR model.

Previous investigations of network resource allocations focused on dynamic problems. One paper that discusses resource allocation on a network is reference 2. This paper considers resources that move across a network, for example passengers in an airplane network. The authors of the paper want to explain a power-law relationship between the strength (amount of resources at a node) and degree of a node. They consider a resource flow with preferential allocation based on the degree of a node. The resources flow around until a steady-state distribution is achieved, similar to that of a random walk. This steady-state has their desired power-law relationship.

Other papers analyze games of competing agents on networks or the flow of resources. Most of them use dynamics to look at their problem and then analyze the resulting steady-state solution (assuming that one exists). On the other hand, for this work, we will not be considering a dynamic process. Our problem is fundamentally different as we want to find the optimal resource allocation given a network.

Disaster management resources are those people, equipment, and vehicles sent into a disaster area to mitigate suffering. For example, these could include tents for refugee camps, stockpiles of vaccines, and trucks for evacuation. These resources have the special characteristic of not being used on a normal basis. They are kept in reserve until a disaster occurs. This characteristic will be useful in creating a mathematical model to describe the optimal resource allocation.

In a disaster situation, the resources need to get to the disaster zone as fast as possible in order to be effective. We will represent this as a decaying benefit for resources that have to travel a distance on the network. This distance will be measured by the number of hops a resource has to travel to get to the disaster site. The edges will be assumed to represent the same real-world distance and to have an infinite capacity for transmitting resources.

The model will have the capability to treat population centers (i.e. the nodes) differently based on their attributes. To keep it simple, we will assume that the nodes are all identical. The probability of suffering a disaster will be the same at every node and the minimum acceptable resources each node needs in a disaster situation will also be assumed to be the same. These two assumptions could be easily relaxed, but they help to keep down the number of adjustable parameters in the model.

Most disasters are unpredictable. We do not know exactly where the disaster is going to occur, although we may have statistical likelihoods. The model will take this into account when optimizing the resource allocation. We will only consider disasters that happen with a uniform probability distribution across the network. Once again in order to keep the model manageable, we will assume that all disasters only affect a single node. This is a serious limitation that is not simple to remove.

The picture to keep in mind is that of a node calling out for help after it is struck by a disaster. All the other nodes in the network (connected to that node) mobilize all of their resources and send them across the network to the disaster site. The resources help in varying degree depending how long they took to get to the disaster site.

The goal of the project is not come out with some detailed model that accurately reflects the real-world. Instead, I merely want to explore resource allocation ideas using a very simple model on a network topology. The real-world problem is incredibly complicated and is certainly not “solved” by the following, but it hopefully sheds some small amount of insight.

Approach / Methods:

Network Creation:

The networks in a real-world application of this idea would have to be constructed from a study of population centers and roads with proper thresholds. It would require another project to figure out a reasonable way to do this. So instead we will use undirected ER random networks produced by Combinatorica. These networks will be entirely suitable to analyze the model’s features. Undirected edges make sense because the grand majority of roads allow distribution in both directions. A possible extension to the project would be to try the model in different network models like the Watts-Strogatz small-world network or Barabasi-Albert scale-free networks.

The Model: (note: vectors are lower-case bold and matrices are upper-case bold)

In order to perform an optimization, we first need to decide on a function that represents the quantity to maximize. In the disaster situation, we want resources to arrive at the disaster site as quickly as possible. So our function should take into account the amount of resources sent by the other nodes and the distances that the resources had to travel. The distances will be computed through the use of the shortest paths matrix $\mathbf{S}$. It has components $\mathbf{S}_{ij}$ which tell us the shortest number of edges that must be traversed to get from node *i* to node *j*. $\mathbf{S}_{ij}$ is defined to be infinity if the two nodes are not connected and to be zero if *i* and *j* are the same node.

Let’s introduce the parameter β ($0 \leq \beta \leq 1$) that will govern the decay over the edges. Each time a resource is sent along an edge it will be multiplied by β. So if the parameter is one, the resource does not decay at all. If the parameter is zero, the resource cannot be sent along the edges. This decay represents that resources need to arrive quickly in order to be effective.

Let's use the vector **r** with component $\mathbf{r}_k$ denoting the resources allocated to node *k* as our resource vector. Then imagine that node *i* is struck by a disaster and examine the following quantity:

$$a_i = \sum_j \beta^{S_{ij}} r_j$$

This quantity tells us the amount of resources that get to node *i* weighted by the decay over the edges for the associated distance.

Now we need to take into account the probabilistic nature of a disaster. Since we do not know which node will be hit by the disaster, one idea is to maximize the average amount of resources that gets to the disaster node as follows:

$$< a >= \sum_i P(i) a_i = \sum_i P(i) \sum_j \beta^{S_{ij}} r_j = \sum_j \sum_i P(i) \beta^{S_{ij}} r_j$$

where P(i) is the probability that node *i* is struck by the disaster. Our assumption is that the probability distribution is uniform, so P(i) = 1/N (N = number of nodes). We can see that the above quantity < a > is unbounded without any constraints. This is unrealistic so we will impose the following constraints:

$$r_j \geq 0 \qquad \forall \qquad j$$

$$\sum_i r_i = 1$$

The first constraint just dictates that the resources are always positive and the second constraint sets a limit on the amount of resources available to distribute. By setting the sum of the resource vector components to one, we can think of the resource vector's components as the percentage of total resources given to each node.

Maximizing < a > under these constraints turns out to be simple. Find the component of the **r** vector with the biggest positive coefficient and put all the resources at that node. Then the value of the optimized < a > is:

$$< a >= \frac{1}{N} \sum_i \beta^{S_{ij}} r_j$$

In some sense, the model is dictating that all the resources are placed at the most central node in terms of distribution.

While this may maximize the average amount of resources < a > over the network, it is not appropriate for a disaster model. Some of the nodes in the network may do very poorly in a disaster while a few nodes may be doing extremely well in a disaster. This situation may have the highest average amount of resources available, but does not seem to be what we want in the optimal distribution of resources.

We do want a high average amount of resources accessible but we also want a minimum amount of resources available at each node. Let μ be the required minimum amount of resources available at each node. Then we now add the following constraint to the model:

$$\sum_j \beta^{S_{ij}} r_j \geq \mu \qquad \forall \qquad i$$

This constraint enforces a mandatory minimum value of resources available at each node. When μ is small, we still expect the centralization of resources we got above, but as μ increases the resources distribute across the network until there is no longer a configuration of resources that can satisfy that minimum level.

So the complete model is as follows:

Maximize (by finding **r**):

$$< a >= \frac{1}{N} \sum_j \sum_i \beta^{S_{ij}} r_j$$

Subject to the following constraints:

$$r_j \geq 0 \qquad \forall \qquad j$$

$$\sum_i r_i = 1$$

$$\sum_j \beta^{S_{ij}} r_j \geq \mu$$

where $0 \leq \beta \leq 1$ and $\mu \geq 0$ are parameters to set based on real-world requirements. The above problem turns out to be solvable through linear programming (using Mathematica). The model is now in a very similar form to the standard OR resource allocation model.

We will explore the behavior of the model in three ways. First we will create a small random network and examine the behavior at a fixed value of β in depth. We will vary μ and examine the resulting resource vector. Method two will consist of creating a slightly larger network and examining the behavior at several values of β. Here we will look at <a> and the distribution of resources using the entropy H. The entropy is defined as follows:

$$H = -\sum_k p_k \log p_k = -\sum_{i=1}^{N} r_i \log r_i$$

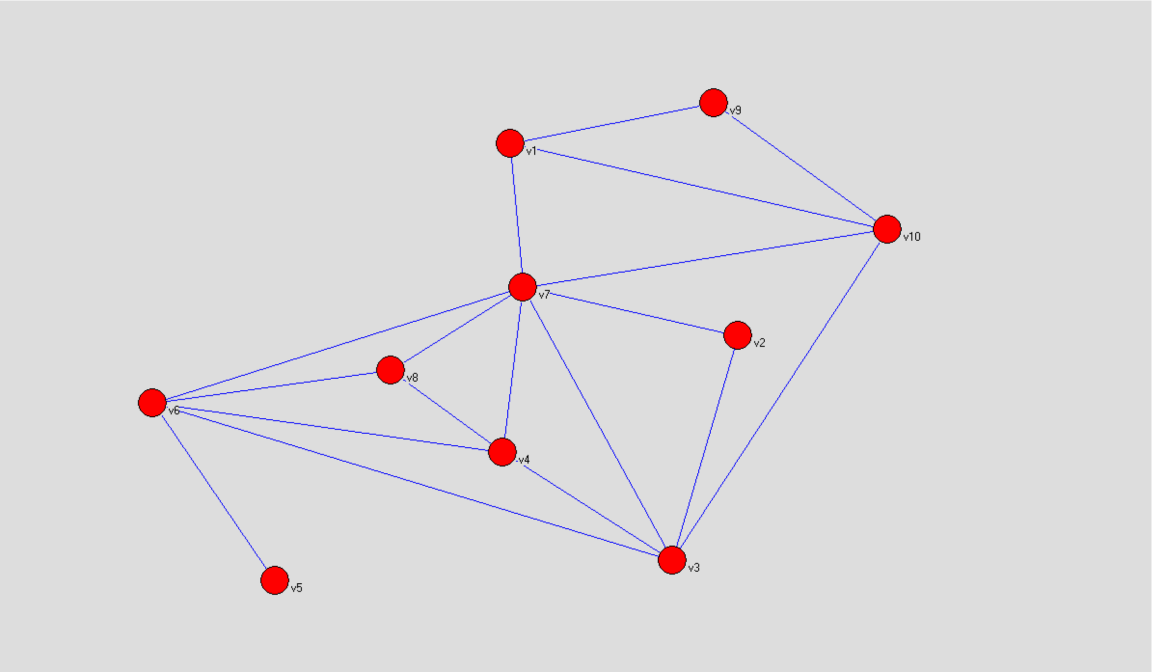
The entropy measures the uniformity of the resource distribution. If all the resources are at one node, the entropy is zero. If the resources are uniformly distributed, the entropy is maximized. The last method will be to check the results against several networks at a fixed beta.

The model suffers when trying to extend it to large networks due to the run-time characteristics of finding all-pairs shortest paths O(N^3) and solving a linear programming problem with N variables. So one problem with the analysis will be that we are not going to be able to figure out how the results scale with system size. An interesting possibility may be to try heuristic allocations based on various centrality measures such as degree or closeness. But even evaluating the performance of one of the heuristic allocations still requires all-pairs shortest paths. To get around this problem, it may be best to institute a cut-off on the length of paths considered based on the value of β. If β is very small, it may only be necessary to consider paths of length one in which case **S** would be very simple to calculate from the adjacency matrix.

Results and Discussion:

Method One:

In order to try out the model, I created a small ten node random network that looks as follows:



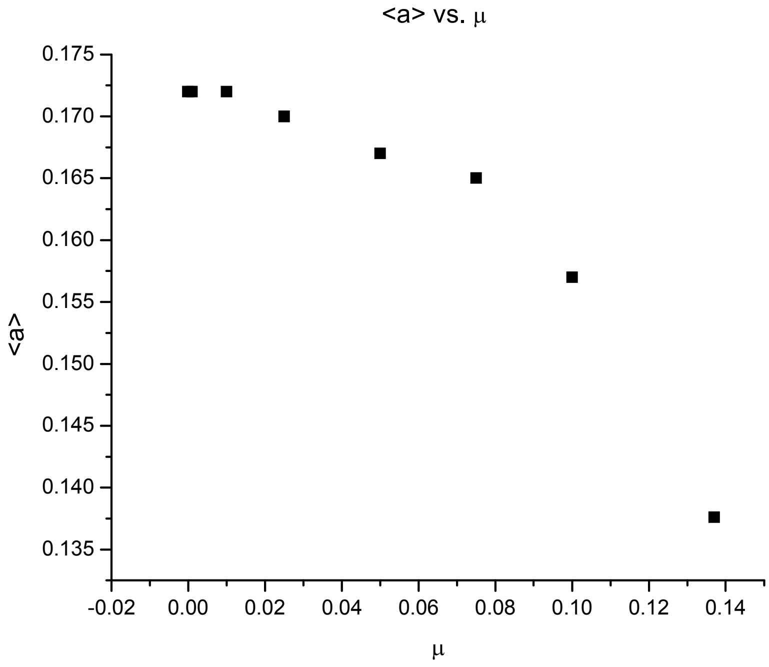
I fixed the value of β as 0.1 and varied μ to find the following.

μ	<a>	Resource Vector
0	0.172	{0, 0, 0, 0, 0, 0, 1, 0, 0, 0}
0.001	0.172	{0, 0, 0, 0, 0, 0, 1, 0, 0, 0}
0.01	0.172	{0, 0, 0, 0, 0, 0, 1, 0, 0, 0}
0.025	0.17	{0, 0, 0, 0, 0.015, 0, 0.969, 0, 0.015, 0}
0.05	0.167	{0, 0, 0, 0, .041, 0, .918, 0, .041, 0}
0.075	0.165	{0, 0, 0, 0, 0.066, 0, 0.867, 0, 0.066, 0.}
0.1	0.157	{.023, .031, .023, .026, .091, .017, .651, .029, .089, .02}
0.137	0.1376	{.102, .114, .084, .096, .124, .084, .085, .10, .11, .094}
0.138		No Solution

Comments:

As expected, we get interesting behavior as we vary the minimum required resources μ. With very low values of μ, it is best to put all the resources on the most important node (7). Looking at the network picture above, we see that seven corresponds to a node that is close (in terms of distance) to every other node in the network. As we increase the minimum required level of coverage, the resource vector begins to change and gradually distribute resources to other nodes. Nodes 5 and 9 are the next to receive resources. Interestingly, 5 and 9 receive resources because they are the farthest away from node 7. They get just enough resources to bring them over the minimum threshold. As the threshold increases, other nodes get more and more of the resources. As we get closer to the point where there is no longer a solution, node 7 actually no longer retains the most resources. The last resource vector has the most resources at node 5 which is interesting because node 5 is “far away” from the rest of the network. These “far away” nodes may become more important at high values of μ because they are the hardest to get resources to. Varying μ changes the picture from one of focusing on the most important centralized node to layering protection over all the nodes.

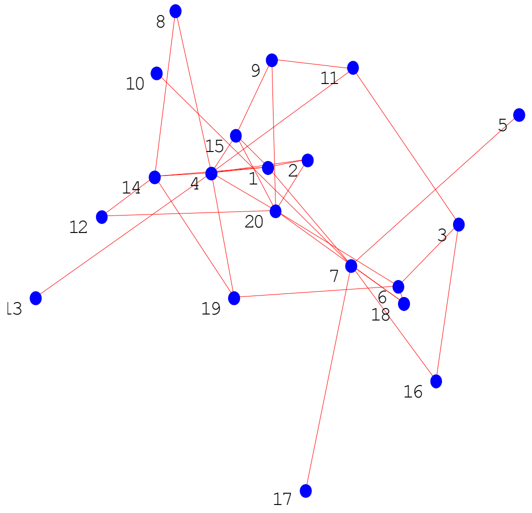
We also should note that the average amount of resources available drops as the minimum threshold is increased. If we graph <a> vs. μ, the decrease does not happen in a linear fashion. For our particular network, it shows that <a> only begins to really suffer as the minimum threshold is pushed up past 0.05.



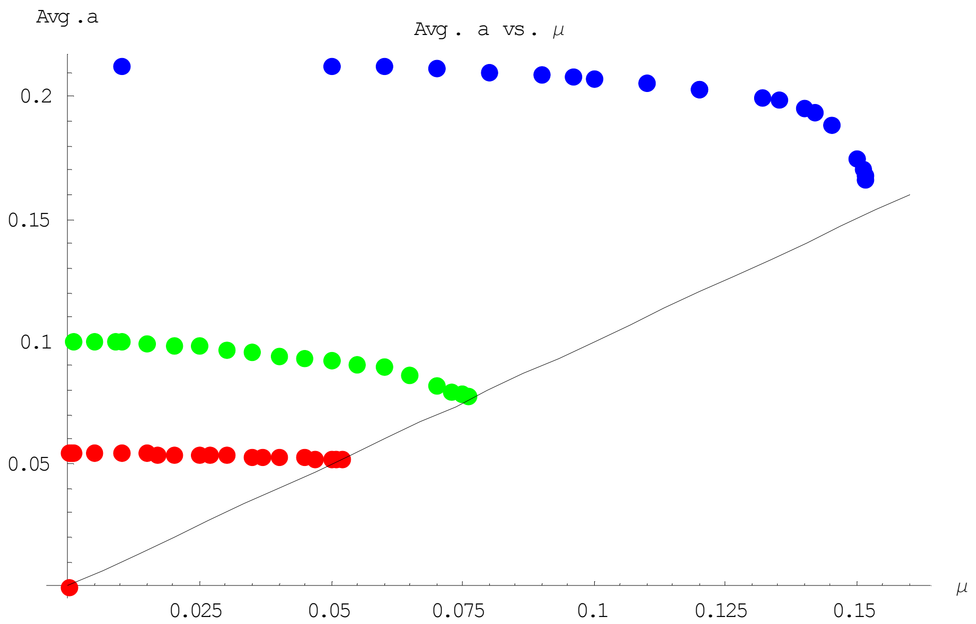
The value of <a> is always greater than or equal to the value of μ, because every node must have at least μ resources available to it. So the value of <a> for any value of μ gives us an upper bound on the maximum value of μ that can give us a solution. As we get closer to the maximum value of μ that allows a solution, <a> should qualitatively approach μ. In the following sections we will observe this behavior for different values of β and different networks.

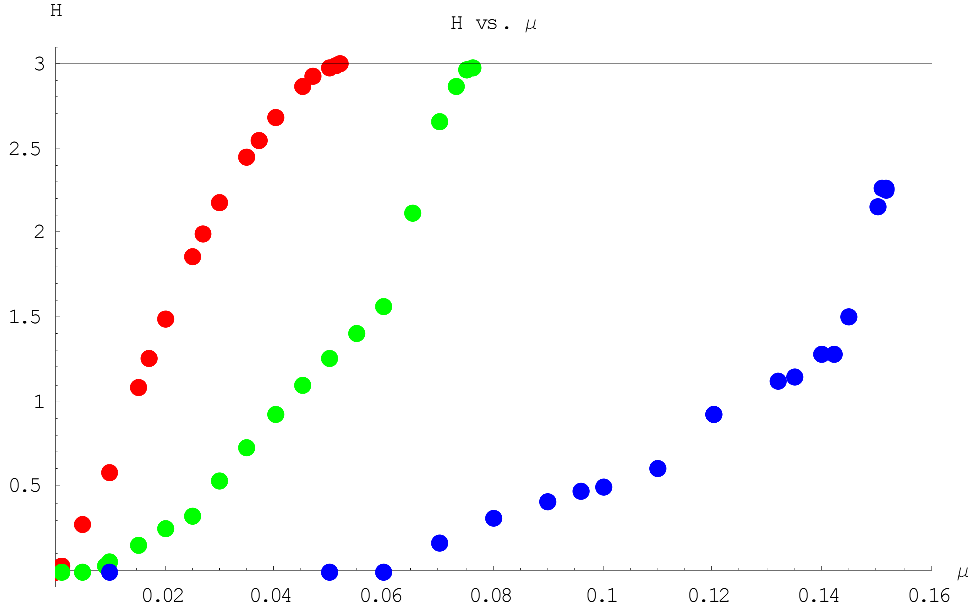
Method Two:

For method two, I created a slightly larger E-R network (N=20, p=.2) using Mathematica.



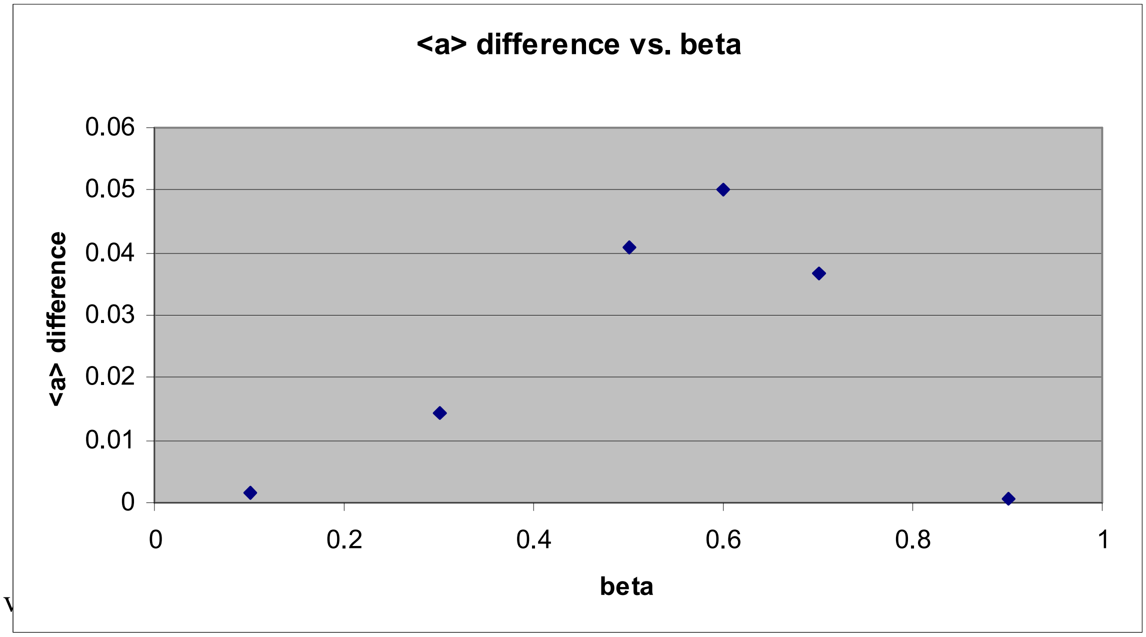
The following graphs for <a> and H vs. μ are color-coded by value of β where red is 0.1, green is 0.3, and blue is 0.5.





Comments:

As expected, with higher values of β the <a> is higher because more resources can travel across the network. The <a> does approach the line <a> = μ as the minimum accessible resources is increased. The variation in <a> appears to depend on the value of β. I made another graph to display this variation, where the difference is from μ = 0 to the maximum value of μ that had a solution for that value of β.



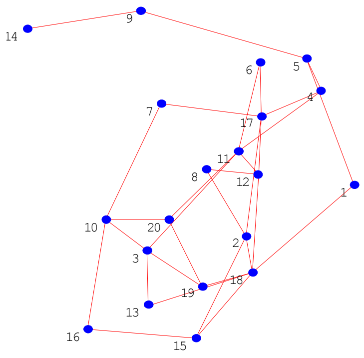
V

An unexpected result shows up in the above graph. The Δ<a> has a peak at a value between zero and one. We can interpret this result by examining what happens at the extreme values of β. When β is very small, the network structure is irrelevant because practically no resources are transmitted. When β is approximately one, the network structure is also irrelevant because the resources can get to any other node (assuming a connected graph) without any difficulty. So somewhere (perhaps at multiple points) between β = 0 and β = 1 must occur places where the effect of the network structure is maximized on the objective function. For the above network this happened near β = 0.6. I suspect that this effect is highly size-dependent.

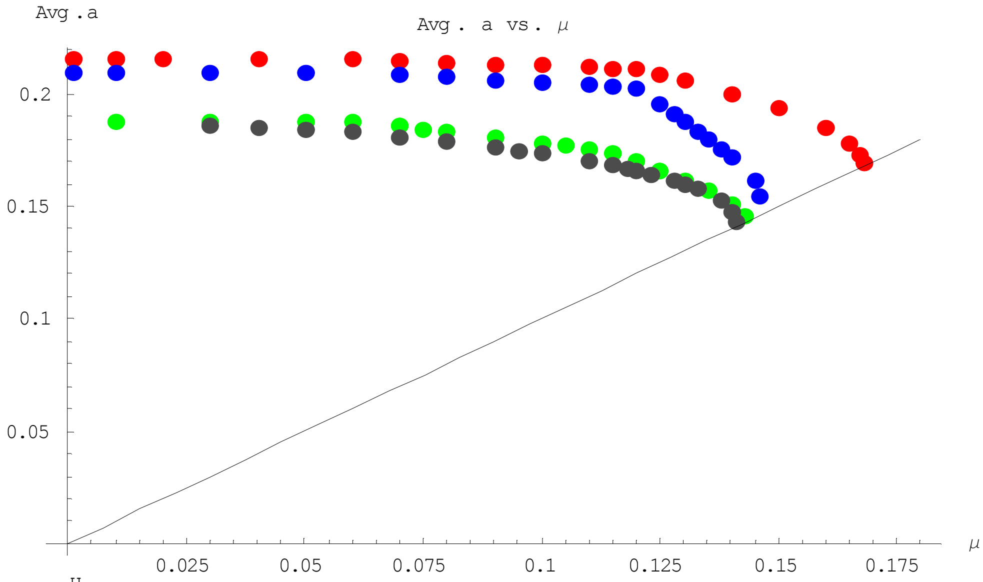
In the graph of H vs. μ, the entropy appears to approach the maximum uniform line drawn at the top of the graph. This means that to get the maximum minimum coverage across the nodes the best distribution is one that is nearly uniform. Method three will confirm that this result was not specific to the given network.

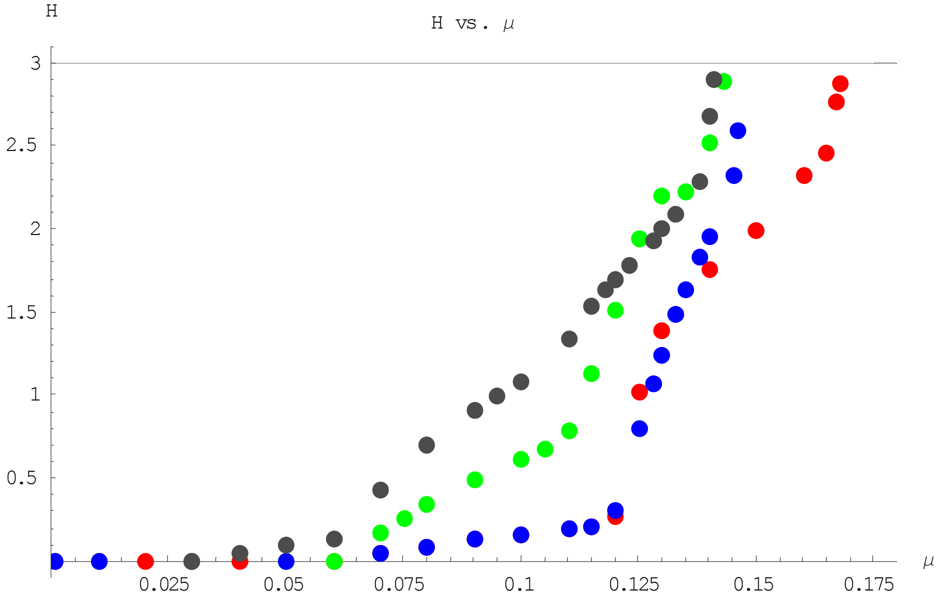
Method Three:

Now I will fix the value of β as .5 and create several E-R networks with N=20 and p=0.2.



Each color in the following graphs represents a different network:





These graphs display the same behavior as the network looked at in the second method. The entropy graph shows a transition from centralized to distributed resources at a value that we can actually predict. The transition occurs based on the node that is farthest away from the "central" distributor solution favored at low μ. The following equation tells us that the critical value of μ comes when the node farthest away is no longer getting enough resources due to the edge decay:

$$\mu_{crit} = \beta^{S_{longest}}$$

This makes intuitive sense and is naturally incorporated into the model.

Conclusion:

The model displays interesting behavior on these small sample networks. The optimal allocation of resources is centralized when the minimum accessible resources required is small. As the minimum accessible resources increases past a critical point dictated by the distances from the central node, the resources distribute across the network. The optimal solution for the maximum minimum accessible resources for which a solution exists turns out to be a nearly uniform resource distribution across the network. The importance of the network structure on the average accessible resources depended on the value of the decay over the edges. With a small decay constant, the network structure is irrelevant to <a> because almost all placements of resources are equivalent. With a high decay constant, the network structure also doesn't matter much because mistaken placement is forgiven by the transfer of resources across the network. A medium value of the decay constant may result in the most difficult real world allocation problems.

Because the μ value tracks the entropy of the resource distribution, it may be appropriate to remove the minimum accessible resource constraint and add an entropy term to the objective function. Now we would be maximizing the average accessible resources and the entropy at the same time with a parameter that dictates the relative importance. As that parameter goes from zero to very large, the resource distribution would change from the centralized distributor to the uniform distribution. This change to the model has the benefit of allowing us to easily find an analytic expression for the resource distribution as a function of the parameters.

The model obviously contains many assumptions that are not acceptable for real-world application, but it remains valuable by capturing some of our intuitive thoughts about resource allocations into a mathematical format. Relaxing the assumptions towards a real-world application will expand the parameter space and likely cause even more complicated behavior.

References:

Operations Research: Models and Methods. 2004. Jensen, Paul A. February 25, 2007. http://www.me.utexas.edu/~jensen/ORMM/models/unit/linear/subunits/resource_allocation/index.html

"Power-Law Strength-Degree Correlation from Resource-Allocation Dynamics on Weighted Networks." Qing Qu, et al. March 01, 2007. http://arxiv.org/PS_cache/physics/pdf/0603/0603081.pdf

"Complex Spatial Organization in a Simple Model of Resource Allocation." Zanette, Damian. March 01, 2007. http://arxiv.org/PS_cache/nlin/pdf/0010/0010037.pdf