Hong Kong University of Science and Technology
Physics Department

PHYS391

Ant Colony Optimization applied to Network Reliability

Anders Bennehag

ID 09515565 : anders@bennehag.com

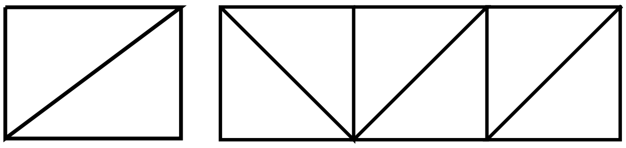
May 16, 2010

Ant Colony Optimization is applied to the design of networks to maximize the reliability of the network if each link has a certain probability to fail. Given a set of possible links and a cost restraint, the algorithm uses swarm intelligence to search areas of the solution space that are likely to contain a good design of the network.

Even though the algorithm is general, one network structure in particular was examined, namely the wheatstone bridge, and chains of this bridge in series. The algorithm finds the globally best design of the 10-cell wheatstone chain after generating 6000 network designs, out of $1.3 \cdot 10^{10}$ possible topologies.

In the latter part of the report, the chains are stacked onto each other to create ring- and cylinder-topologies and the reliability dependence on the dimensions of these objects is determined. Furthermore, how many networks can be constructed from a given number of links is investigated.

Contents



(a) (b)

Figure 1: (a) A typical wheatstone cell. (b) A chain of three cells.

1 Introduction

1.1 Network Reliability

The problem of network reliability or determining the reliability of a network is crucial to many areas like energy distribution, communication-networks, and traffic-control. Given a set of nodes and connecting edges with a probability to fail, what is the probability that two chosen nodes are connected through a subset of edges? Or the probability that all nodes are connected? These probabilities are called the two-terminal and all-terminal reliabilities of the network, respectively.

The problem of finding a network design that maximizes reliability and/or minimizes cost belongs to the class of NP-hard problems. That is, the search space grows exponentially with the amount of nodes. Therefore, large instances of this problem cannot be solved analytically nor by exact computational methods. But optimization algorithms where close-to-optimal solutions are generated can be used. Examples of these algorithms are genetic algorithms[1] and ant colony optimization[2].

The purpose of this paper is to investigate the reliability of a special network structure called the wheatstone bridge, pictured in figure 1a. It is of interest to determine how a chain of these bridges performs and how the placement of the diagonal links affect performance, see figure 1b.

1.2 Approximate solution methods

Optimization problems are characterized by the relative complexity of the problems which makes them hard to optimize. Many of these problems belong to the group of NP-Hard problems, which means that the time it takes to find the guaranteed global solution grows exponentially or faster with the size of the problem. This makes exhaustive search and other exact solution methods useless on real-world sized problems.

This has created a need for approximate solution methods that may or may not find the best solution. Many of the new solution methods are robust and are being applied in very diverse fields. One of the most famous methods is genetic algorithms where a solution is encoded as a binary string which symbolizes the chromosomes. Better solutions are found by mutating and cross-fertilizing good solutions. Another example is the particle swarm optimization[3], where particles describing different solutions are given random velocities in the solution space. By letting the best solution or particle attract other solutions, a more thorough search of that area in the solution space is performed.

The ant colony optimization, ACO, can be seen as a version of the particle swarm for discrete-valued solution problems. Where the particle swarm searches the entire real-valued solution space, the ant colony excels at finding solutions to combinatorial problems, for example the travelling salesman problem, or time scheduling problems. The ant colony optimization mimicks the swarm intelligence observed in ant colonies. As one ant find food, it creates a trail of pheromones back to the colony. Other ants might follow these trails and shorter paths results in more pheromones being deposited. After some time, the path with the most pheromones is likely to be a good approximation of the best path.

What is common for all these algorithms is that they often generate satisfying solutions but the algorithms rely on stochastic processes and there are no proofs of how good the solutions actually are. But experience has shown that they find very good solutions to NP-Hard problems in general.

2 ACO applied to network reliability

2.1 Introduction

One of the goals when designing (reliable) networks is to maximize the reliability given certain cost restraints. Given a set of possible links and a cost restraint, what topology or design will maximize the reliability of the network, given that each link has a probability to fail?

The ant colony algorithm is applied to the design of reliable networks by letting an ant represent one design. Ants that find more reliable designs are allowed to deposit pheromones onto the links they chose. These pheromones will affect the decisions of future ants when they choose links to traverse the network. Links with higher levels of pheromones have been part of successful designs in earlier generations and new ants should therefore be more likely to include these in their path.

2.2 Algorithm

Given a network with all possible links, the ant colony optimization has the objective of maximizing the reliability without breaking any cost restraints.

The algorithm is initialized by associating an initial pheromone level $\tau_{ij}(0) = \tau_0$ with every link ij. After this, A solutions are generated by letting each ant a traverse the network and pick paths according to a probability function $p_{ij}(t)$, equation 1. This probability is a function of the pheromones deposited by earlier generations of ants.

The best solution in terms of reliability without breaking the cost restraint is then used for the global updating rule for distributing new pheromones, equation 2. This process of generating new ants and depositing pheromones is iterated until some stopping criteria has been fulfilled, in our case, K iterations are performed.

The algorithm can be summed up as

```
Set pheromone levels to initial state
While continuing:
        Generate A solutions
        Estimate the reliability of each solution
        Update the pheromone levels
        Keep the best ant and remove the rest
        Iterate (K times)
```

The syntax for describing links is ij, which means level j of link i. In general, it is possible to have different types of links with for example; a smaller probability to fail at a greater cost. In our problem, there is only one type of link which means that $j \in (0, 1)$ where $j = 0$ means no link and $j = 1$ means that there is a link.

The probability p_{ij} of choosing a particular link ij is defined as

$$p_{ij}(t) \equiv \frac{|\tau_{ij}(t)|^{\alpha}|\eta_{ij}|^{\beta}}{\sum_p |\tau_{ip}(t)|^{\alpha}|\eta_{ip}|^{\beta}} \tag{1}$$

τ_{ij} and η_{ij} is the pheromone level and cost, respectively, over link ij. In our problem, the cost is constant for all the links and the term can be disregarded. α and β are constants to weigh the two factors against each other. To normalize; the the denominator is the sum over all the connecting links.

The level of pheromones over link ij is given by equation 2 and is called the global updating rule.

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{2}$$

$\rho \in [0, 1]$ is the evaporation coefficient that determines how fast older trails disappear. $\Delta\tau_{ij}$ is the sum of all the pheromones of ants passing over link ij as follows:

$$\Delta\tau_{ij}(t+1) = \sum_a \Delta\tau_{ij}^k(t+1) \tag{3}$$

$$\Delta\tau_{ij}^k(t+1) = \begin{cases} \left(\frac{R_k}{R^*}\right)^b & \text{if ant } k \text{ passes over } ij \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

R_k is the reliability of the kth ant and R^* is the reliability of the best found ant.

As it turns out, the speed of convergence is highly dependant on the exponent b. The fraction R_k/R^* is often very close to 1 which means that b must be very large in order to distinguish good solutions from bad. For example, in the 10-cell chain, it seemed to work well with $b = 100$. An idea is to dynamically adjust b during execution of the algorithm when there are estimates of the fraction R/R^*.

2.3 Improvements

As time evolves, the ants are likely to find local minimas but there are improvements to the algorithm which decrease this risk of entrapment which have proven to give good results [4].

One such improvement is to apply a neighborhood search of the best solution in each iteration meaning that each link in the solution is rearranged one at a time to see if a better solution is found.

Another solution is to reset the pheromone levels and start over if the same solution is found over several iterations.

2.4 Monte Carlo-simulation

As the size of the network grows larger, exact solutions become intractable. Therefore it's necessary to find approximate solutions and one such method is a simple Monte Carlo-simulation [5].

The reliability is calculated by simply generating many instances of the network where each link fails with a given probability and examining if all nodes can be reached. The reliability of the network, R, is then given by

$$P = \frac{\text{instances where all nodes are connected}}{\text{all instances}} \tag{5}$$

Of course, it is important to do many simulations in order to get a reliable estimate of R.

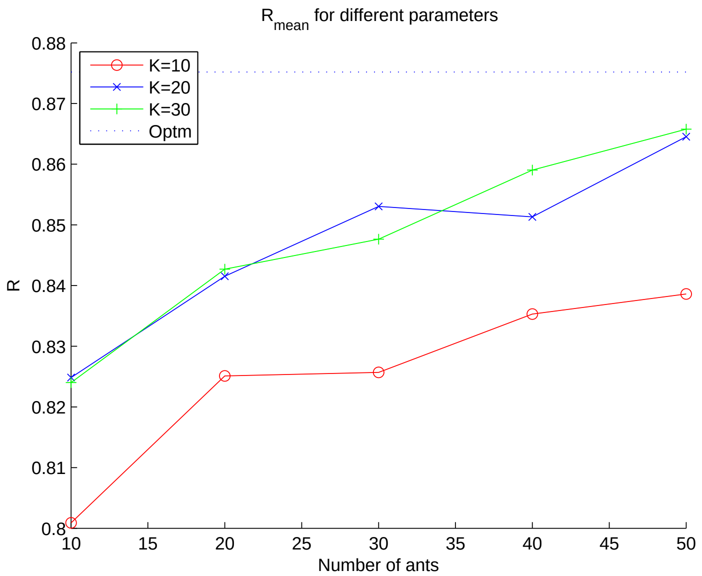
2.5 Results

The algorithm has been tested on the one dimensional wheatstone chain, consisting of 10 cells. Figure 3 shows a 4-cell wheatstone chain. The 10-cell chain has 51 possible links, and the cost restraint is to choose 41 of those. Therefore there exists $\binom{51}{41} \approx 1.3 \cdot 10^{10}$ possible topologies. There's no possibility to examine all of these by exhaustive search.

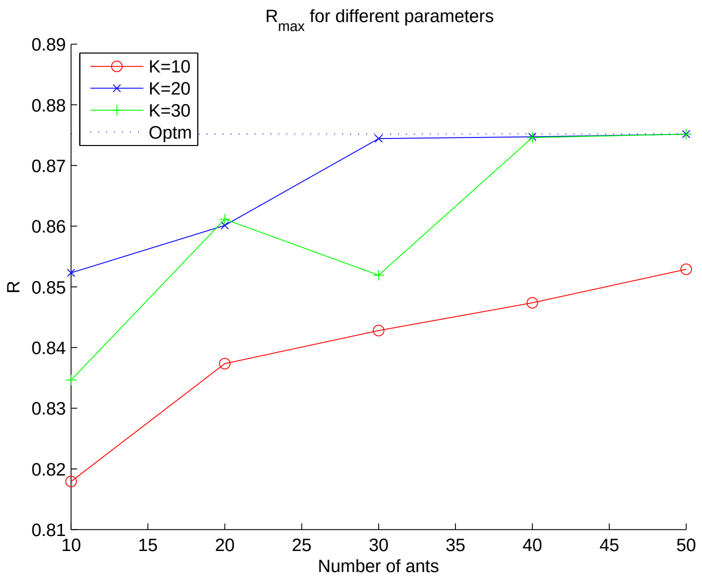
Figure 2a shows the averaged results from 10 runs of the ant colony algorithm for different values of the parameters. The parameters that are being adjusted is the number of ants in each generation, and the number of generations that are created before the algorithm ends. It is obvious that generating 30 generations of ants instead of 20 gives no further gain, which implies the bad links have a very low probability to be chosen already at $K = 20$. When this point is reached, where there are no further improvements, is very problem specific, but it would be simple to abort the algorithm when the same solution has occured enough times, or when a good enough solution is obtained. As the number of ants in each generation is increased, there is a corresponding increase in the average R_{max}.

Figure 2b shows the best R that was found over 10 runs of the algorithm. This gives that the global maximum is found already when using 30 ants and 20 generations. Each run of the algorithm therefore tested $20 \cdot 30 = 600$ solutions. The algorithm found the globally best topology after running 10 times and testing 6000 possible topologies out of $1.3 \cdot 10^{10}$.

Table 1 shows the behaviour of the algorithm when b (see equation 3) is changed. b is the exponent to the fraction describing how much pheromones to drop off on each link relative to the best solution.



(a)



(b)

Figure 2: (a) The examined network is the 10 cell wheatstone chain. Datapoints show the reliability of the solution found by ACO averaged over 10 runs, for different values of K and ants a. The best solution has a reliability of 0.874 and is described by the dotted line. (b) Same as figure 2a but datapoints now show the best reliability as found by ACO over 10 runs, for different values of K and ants a.

Table 1: Mean and standard deviation for 5 runs of ACO with the following parameters. The number of tested solutions equals to $N_{\max} \cdot \text{Ants} = 600$

$N_{\max}$	Ants	b	R_{mean}	R_{stddev}
30	20	20	0.843	0.020
		50	0.864	0.017
		100	0.882	0.017
		500	0.876	0.017
		1000	0.877	0.018

The convergence speed towards a good solution depends on b in the sense that larger b gives better solutions. In reality this gives the effect that only the best found solution deposits pheromones. Further testing will be done to verify this. If only the best solution deposits pheromones, it means that as older trails evaporates, the search space will be concentrated to topologies more and more similar to the best solution so far.

One explanation as to why it's best to only let the best solution deposit pheromones might be that the reliability R is estimated and not calculated exactly. This means that it might be hard to differentiate good solutions from bad in equation 3. Since the fraction might be very close to 1 bad solutions could deposit approximately as much pheromones as the best solution, which would inhibit convergence.

2.5.1 Preferred structure of the wheatstone bridge

When applying the ant colony optimization to the one dimensional chain it shows that the preferred structure is the one of figure 3. The vertical links inside the chain often have very low amounts of pheromones which implies that they are less important for the overall reliability of the network. The vertical links at the ends of the chain have more pheromones instead, they are overall more important considering the four courner nodes have fewer connecting links.

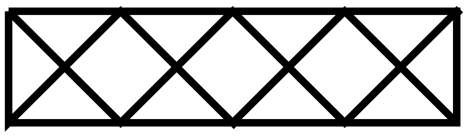


Figure 3: The preferred structure of the 4-cell network as determined by the ant colony optimization.

3 Investigation of different chains and cylinders

3.1 Introduction

After finding the best topology for the wheatstone bridge we proceed and examine how the chain of wheatstone bridges can be stacked onto each other connected into rings and cylinders. In the next subsection the terminology describing the dimensions of these structures is established and then we examine how the dimensions affect reliability.

3.2 Notation

The width of a chain is defined as the number of cells, or spins in the earlier notation, that the chain consists of. For example, the chain in figure 3 has four cells and is therefore of width 4. The height of a chain is simply the number of 1d-chains stacked on top of each other. As an example, figure 4 sports a 4 cells wide and 2 cells high chain-topology.

A ring- or cylinder-topology is a chain where one end of the chain is connected to the other end.

With this terminology established, lets examine how the reliability depends on the dimensionality of the network.

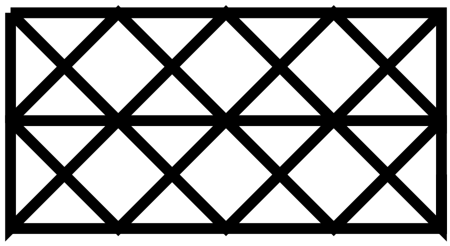


Figure 4: The topology of a chain with width 4 and height 2.

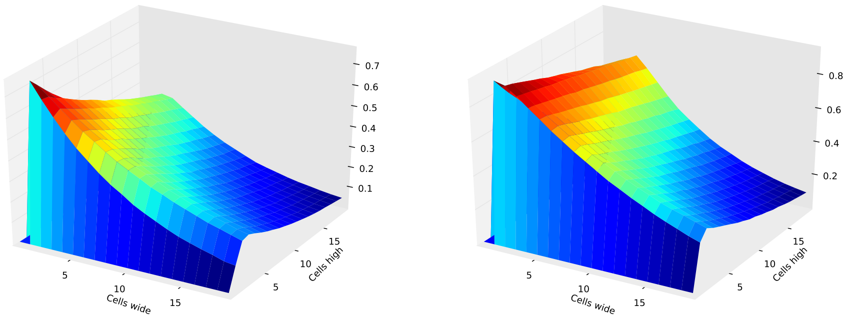
3.3 Method

How the reliability depends on the dimensions of the network was examined by generating networks with different width and height and then estimating the reliability with Monte Carlo-simulation.

3.4 Results

The results are displayed in figure 5a and 5b respectively. Increasing the height of the network results in a slower decrease of the reliability as compared to increasing the width. This is because an increase in height results in more edges being added to the network than the same increase in width.

What is more interesting is that the Ring-topology, figure 5b, is much more independant of height than the Chain-topology, figure 5a without having more links.



(a) (b)

Figure 5: (a) Reliability of a Chain-topology. How different dimensions of the chain affects the reliability. (b) Reliability of a Ring-topology.

4 Degeneracy in regards to link density $\rho(l)$

4.1 Introduction

It might be interesting to investigate how many possible networks can be constructed from a certain concentration $\rho(l)$ of links, in other words, the degeneracy $D = D(\rho(l))$. $\rho = 0$ means that the network consists of no links and $\rho = 1$ means that it is fully connected.

The real problem was how to determine if a network is symmetric to some already discovered network. I have managed to detect rotational symmetries if we assume that the nodes are places in a circle as in figure 6. Detection of symmetries that arise when mirroring the network is possible, but requires additional work and will not change the results as is discussed in the results.

4.2 Algorithm

The complexity of the algorithm, see equation 6, is a factorial of the number of links, namely $\binom{l_{\max}}{l}$ which imposes serious restrictions on the size of the problem. The largest network that is solvable has 7 nodes. The n^3 terms in the complexity comes from checking against symmetries but that hardly changes the situation from a complexity point of view.

$$C \in O\left(\binom{N(N-1)/2}{l} \cdot N^3\right) \tag{6}$$

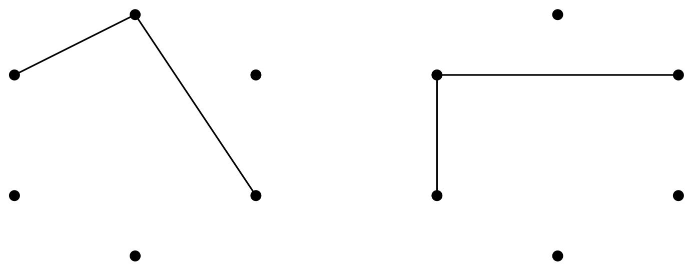


Figure 6: Two topologies of a 6-node network that are rotationally symmetric.

4.3 Results

The results for a 7 node, circular network with and without rotational symmetries can be found in figure 7. It's not the most interesting result as there's only one maxima.

Removing the mirrored symmetries will not change the results for the same reason removing rotational symmetries don't change the results. There can only be $N-1$ symmetrical topologies in a rotational sense because there can only be $N-1$ shifts of the links before the first topology returns. This is why the data describing all of the topologies in figure 7 is a factor 7 larger than the data describing the topologies where the rotational symmetric networks have been removed.

By the same argument, there can only be $N/2$ mirrored symmetrical networks of one particular topology.

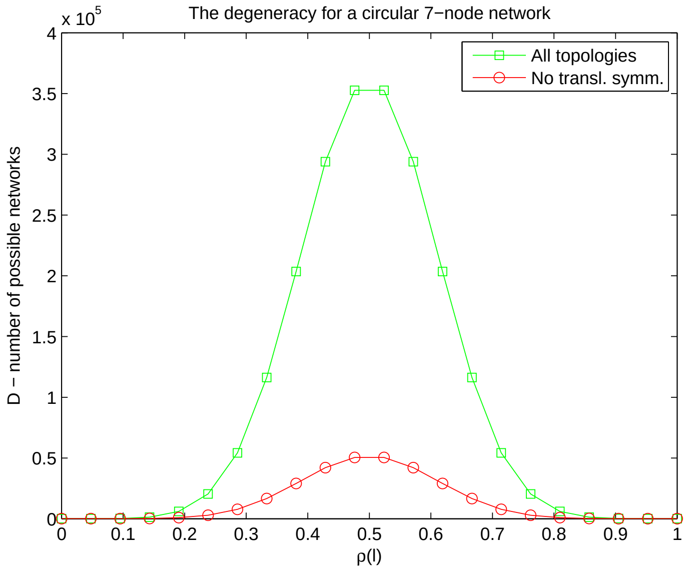


Figure 7: How the degeneracy, or number of possible networks, depends on the link density $\rho(l)$ for a 7-node circular network. For the squared line, all generated networks are counted. For the circle line, rotational symmetries are not counted, but mirrored symmetries are.

References

[1] David Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning*, page 41. Addison Wesley, 1989.

[2] Marco Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, 1992.

[3] James Kennedy and Russel Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks IV*, 1995.

[4] Kanyapat Watcharasitthiwat and Paramote Wardkein. Reliability optimization of topology communication network design using an improved ant colony optimization. *Computers & Electrical Engineering*, 35(5):730 – 747, 2009.

[5] George S. Fishman. A comparison of four monte carlo methods for estimating the probability of s-t connectedness. *IEEE Trans. Reliab*, 145(2), 1986.