ABSTRACT
In this paper, we propose an Ant Colony Optimization algorithm for the network inference and parameter estimation of S-Systems. The ACO has been used for various problems, and with several improvements, it can also be used to the problems that we are considering.

Keywords
Parameter Estimation, Network Inference, Local Minimum, S-Systems

1. INTRODUCTION
The use of Biochemical Systems Theory (BST) has found many applications in Metabolic Engineering, Drug development, etc. The GMA and S-Systems formulations of BST are non linear differential equations which involves parameters whose values have to be estimated. Unlike linear systems where effective parameter estimation can be used, there is currently no available algorithm which can handle S-System or GMA.

In this study, we would be solving two problems: the Network Inference and the Parameter Estimation problems. Network inference involves determining which chemical species interact within the biochemical network, and this is usually done using some fit to data. The Network inference problem can be considered as a combinatorial optimization problem, since the goal is to determine whether a certain metabolite is involved in a reaction or not.

The parameter estimation problem involves determining the actual parameter values of the model, and thus requires a real-valued algorithm. With this problem, we propose to use a continuous variation of the Ant Colony Optimization algorithm.

It is known that ants have the behavior of leaving a certain kind of chemical called as pheromones. This chemical acts as a tool for the ants to communicate, so that they’ll know which paths are being used by the other ants. The more pheromone concentrated in the area implies that there were more ants which passed by the area, and hence it should be a better route for them to find their food. This idea was the one used by Marco Dorigo, in developing a novel optimization algorithm now known as the Ant Colony Optimization Algorithm.

Several modeling frameworks are being used in the analysis of Biochemical pathways. Some models involve the use of polynomial models, while some uses Artificial Neural Networks. In this paper, we concentrate on S-Systems model. This model involves two kinds of parameters, kinetic orders, usually represented by g and h, and rate constants, usually written as: α and β. This parameters map one to one onto the structure of the metabolic network. Therefore the problem of determining the structure of the model just depends on the estimation of the parameters.

We now look at the structure of an S-system based biochemical model.

Given n biochemical species, with concentration at time t denoted with: \(X_1(t),...X_n(t)\), the biochemical model can be represented as a system of ordinary differential equations of the form.

\[
\frac{dX_i(t)}{dt} = \alpha \prod_{j=1}^{s_i} X_j^{r_{ij}} - \beta \prod_{j=1}^{s_{hi}} X_j^{h_{ij}}
\]

Each differential equation in the system can be viewed as a model representing a single reaction. The system of differential equation on the other hand represents a biochemical network consisting of a series of reactions. Given a differential equation, with a metabolite having a 0 kinetic order implies that the metabolite is not involved in the reaction.

2. PROBLEM
The main problem that this research is trying to solve is on how to effectively produce the model given the concentration of the metabolites. We would want to find the set of parameters (kinetic orders and rate constants) that would produce the given concentrations. This is considered as an inverse problem, because we are given the results of a model, and what we need to find is the structure of the model.

This inverse problem is a difficult problem since we need to estimate a large number of parameters, given only the time series of the concentration of metabolites. In general, given a set of n metabolites, we need to estimate a 2n rate constants and 2n^2 kinetic orders.
3. RELATED LITERATURE

The problem of finding the corresponding structure and formulation of an S-System given the concentration of metabolites, has generated interest of scientists all around the world. As a result, a group of researchers from UP Diliman, and Munich Germany has formed a weekly seminar group that aims to create a benchmarking framework in solving biochemical systems. This group is collectively known as the MAD [6].

In their paper, they presented several methods that were used in solving the network inference and parameter estimation problems. Among the methods that were used are: Particle Swarm Optimization (P.Naval), Simulated Annealing (O.Gonzales, M. Echavez), Newton Flow (M.de Paz, R. del Rosario), Genetic Algorithm (M. Bargo). All the methods mentioned, and many of the other methods that were previously used are stochastic methods. This is because to the fact that the hardness of the problem has made it impossible for a deterministic algorithm to solve the problem.

The results produced by the various algorithms are almost synonymous. They were able to solve the same networks with accuracy. (HA 96), but almost all of the methods failed on the CM06 network.

4. ANT COLONY OPTIMIZATION

4.1 Discrete ACO Algorithm

The Discrete Ant Colony Optimization Algorithm can be characterized by two things:
- The probabilistic transition rule to determine the direction of each of the ants.
- The pheromone update mechanism.

In solving the Traveling Salesman Problem, the algorithm considers a set of n solutions (called as ants) and performs several iterations until a certain termination condition is reached. For the TSP, a solution to the problem is any random tour that does not pass any city more than once and that terminates at the starting point. For the first iteration, the probability that a node will be selected is just the same as the other nodes. While making a tour, each ant will leave pheromone trails on its path. The amount of pheromone that will be left in the path will depend on the distance of the path traveled. The shorter the distance, the more the pheromone that will be left, while the longer the distance, the lesser the amount of pheromone that will be dropped. For the succeeding iterations, the selection of the nodes will depend on the amount of pheromone that was left in the trail. The more pheromone that were left the higher the probability that a certain path will be chosen.

The amount of pheromone that will be left on ground will be updated depending on the cost of value of the solution. Consider a general optimization problem where, given a set of discrete choices, the goal is to find the best combination of choices depending on a given fitness function. Usually the fitness function is the cost function between the true solution, and the solution generated by the algorithm.

4.2 Continuous ACO Algorithm

A similar approach is used for solving continuous optimization problems. This time we are given a continuous fitness function, \( f(x_1, \ldots, x_n) \). The objective is to find the vector \( x \) that will minimize (or sometimes maximize) the fitness function. We have to note, that the elements of the \( x \) are real numbers.

The continuous ACO algorithm solves this problem by first generating a set of \( m \) random vectors. These solutions were generated from a multivariate Gaussian distribution. The initial mean and the variance of the multivariate Gaussian distribution is arbitrary.

After the generation of each vectors, the fitness of each vectors will be computed and the resulting fitness values will be sorted. The vector that has the best fitness value will then be used for the next iteration of algorithm.

New solutions will be generated in the next iterations, but this time, instead of using an arbitrary mean for the multivariate Gaussian distribution, the best vector that was generated in the previous algorithm will be used. The process will be repeated, until a certain number of iterations are used, or until a certain threshold value is obtained.

5. PROPOSED ALGORITHM FOR S-SYSTEMS

5.1 Network Inference

We propose to use the discrete ACO algorithm for the network inference problem. The network inference problem involves the selection of the metabolites that are involved in each of the reaction in the S-System. We made some improvements in the discrete ACO algorithm to make it fit for the network inference problem.

In the proposed algorithm, instead of choosing choices once at a time, for each solution, we selected the choices at once, and computed for its fitness value. This was done since the order of selection of metabolites is not material to the problem. Also, to ensure that proportional amount of pheromones will be assigned for each possible solution, we used the following function to assign pheromones

\[
P(X) = \frac{1}{F(X)+1}
\]
Where, $F$ is the fitness function. In the equation, the higher the fitness value, the lesser the amount of pheromone that will be left for the solution. It is easy to notice, that the function would limit the distribution of pheromone between the range $0 – 1$.

<table>
<thead>
<tr>
<th>Select nodes randomly</th>
</tr>
</thead>
<tbody>
<tr>
<td>While termination condition is false</td>
</tr>
<tr>
<td>For each ant</td>
</tr>
<tr>
<td>Construct solution according to pheromone values</td>
</tr>
<tr>
<td>Compute fitness value of the constructed solution</td>
</tr>
<tr>
<td>Assign pheromone values based on fitness value.</td>
</tr>
<tr>
<td>Use fitness value equation.</td>
</tr>
</tbody>
</table>

Figure 1: Discrete ACO for Network Inference.

The fitness function used in this research is the relative error of the resulting model’s output (based on the results of the algorithm) as compared to the model’s output based on the true parameter value. If $I$ is the output based on the algorithm, and $I'$ is the output based from using the true value, then the fitness function is given by:

$$F = \frac{\|I - I'\|}{\|I'\|}$$

### 5.2 Parameter Estimation

We propose to use the ACO algorithm for continuous problems in the parameter estimation problem. But several innovations were made to ensure that the algorithm will be useful for S-Systems.

1. We used the same process of distributing pheromones as the one that we used in the discrete case. This function was used to ensure the normality in the amount of pheromone that is used.

2. Use the fitness value to determine the variance that will be used in the multivariate Gaussian distribution. This step is done to allow the variance to converge to 0 if the fitness value is low enough. On the other hand, it will also allow the variance to be high for high fitness values. This is helpful so that the Gaussian distribution would be allowed to have a wider variance during cases when the fitness value is high.

The second innovation is one of the most critical step in this algorithm. The challenge right now is to find a suitable relationship between the fitness value and the variance.

Consider an S-System model with the rate constants $A$ and kinetic parameters $B$.

1) Generate random values for the said set of parameters.
2) Compute for the resulting cost of the said randomly generated parameters.
3) Store in a table
4) Repeat steps 1 – 4 for $m$ times.
5) Sort the table based on the cost values
6) From the table, randomly select new parameters $A$ and $B$. The selection should be biased on the parameters with lower costs
7) Use the selected parameters as mean of a multivariate Gaussian distribution.
8) Generate new solutions from the distribution
9) Compute for the cost
10) Sort the table
11) Update the variance
12) Repeat steps 6 – 11 until the terminating condition is satisfied.

Figure 2: Continuous ACO for Parameter Estimation Problems

### 5.3 Jumping Ants

The problem of falling on local minimas is a common problem in parameter estimation. The same is true for the parameter estimation of S-Systems. In this research, we propose the use of Jumping Ants. In the algorithm, if the variance starts converging to 0, then it means that the algorithm is approaching a solution. If the resulting fitness value is still high enough, then it means that the algorithm is converging to a local minimum, rather than in a global minimum. If such a condition exists the algorithm should automatically set the variance to a larger constant $c$, to allow the Gaussian distribution to choose a solution that is significantly farther than the current solution. In this case, the algorithm will start finding other solution other than the current solution.

### 6. RESULTS

Currently, we are at the process of developing the discrete ACO algorithm for the network inference of biochemical networks. We have already applied the continuous algorithm for the VA04 network. The researchers are currently looking at how the variance will be converging so that it can converge to a solution, and yet allow variability so that it the algorithm can converge to the best solution possible.

The VA04 network is a network consists of 1 independent variable ($X_0$) and 4 dependent variables.
\[ \frac{dX_1}{dt} = 20X_0(t)X_1(t)^{-0.8} - 10X_1(t)^{-0.8} \]
\[ \frac{dX_2}{dt} = 8X_1(t)^{0.5} - 3X_1(t)^{0.75} \]
\[ \frac{dX_3}{dt} = 3X_3(t)^{0.75} - 5X_3(t)^{0.5}X_4(t)^{0.2} \]
\[ \frac{dX_4}{dt} = 2X_1(t)^{0.5} - 6X_4(t)^{0.8} \]

We used the fitness function that was presented above. In our experiments, we considered 2 cases:

Case 1: Variance converges independently from the fitness value
Case 2: Variance converges with the fitness value.

For Case 1, we let the variance converge independently. We used two methods to reduce the variance. The first method is by subtracting a constant from the variance. The second method is done by multiplying a certain constant on the variance. For the first method, it is seen that fitness function converges too slowly, thus allowing the algorithm to find other possible solutions.

The second method on the other hand makes the variance converges to 0 too quickly. In this case, the algorithm always produces a local minimum, since the algorithm finds a solution; it would be converging to the solution.

For case 2, the variance was decreased by multiplying it to the fitness value of the past iterations. This was done so that the variance becomes dependent on the fitness value. A high fitness value would make the variance larger, hence the algorithm would be more open to other solutions. A low fitness value on the other hand makes the variance smaller, hence the algorithm will be converging to the solution producing the low fitness value.

The results have shown that the fitness value was oscillating while converging. This is due to the fact the variance depends on the fitness value. If the fitness value is high then the variance will also be high, hence more variation. While if the fitness value is low, then so is the variance.

Another thing that can be observed is that despite its oscillatory behavior, the magnitude of the fitness value steadily decreases, implying convergence of the solution that is produced by the algorithm.

7. CONCLUSION

The researchers are currently on the process on fine tuning the ACO so that it can be used for the network inference and parameter estimation of S-Systems. The researcher plans to use the solution on the network inference algorithm to lessen the search space for the parameter estimation algorithm.
The results of the experiments have shown that the possibility of using the fitness value as a determinant for the variance.

The ACO algorithm can be used on the network inference and parameter estimation of S-Systems. The discrete ACO algorithm can be used on the network inference problem, while the continuous ACO algorithm will be used for the parameter estimation problem. Several innovations are implemented to make the said algorithms suitable for S-Systems.

8. REFERENCES


9. AUTHOR’S AFFILIATION

1) Department of Computer Science, University of the Philippines – Diliman

2) Marine Science Institute, University of the Philippines - Diliman