University of New Mexico
UNM Digital Repository

Computer Science ETDs | Engineering ETDs

7-1-2011

Improving peer review with ACORN : Ant Colony Optimization algorithm for Reviewer's Network

Mark Flynn

Follow this and additional works at: https://digitalrepository.unm.edu/cs_etds

Recommended Citation
Flynn, Mark. "Improving peer review with ACORN : Ant Colony Optimization algorithm for Reviewer's Network." (2011).
https://digitalrepository.unm.edu/cs_etds/58

This Thesis is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Computer Science ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.

Mark Flynn
Candidate

Computer Science
Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

_____, Chairperson

_____

_____

_____

_____

_____

_____

_____

_____

IMPROVING PEER REVIEW WITH ACORN: ANT COLONY OPTIMIZATION ALGORITHMS FOR REVIEWER'S NETWORK

by

MARK FLYNN

B.A., BIOLOGY AND PSYCHOLOGY
UNIVERSITY OF DELAWARE, 1992
PH.D., NEUROBIOLOGY
UNIVERSITY OF DELAWARE, 2001

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Computer Science

The University of New Mexico
Albuquerque, New Mexico

July, 2011

ACKNOWLEDGEMENTS

I gratefully acknowledge the assistance and help I have received throughout my time as a CS graduate student at UNM. I thank all of the members of the Scalenet lab, Tatiana Paz Flanagan, Kim Kanigel-Winner, Kenneth Letendre, Josh Hecker, Soumya Banerjee, Neal Holtschulte and Thanaphon Tangchoopong, for their helpful comments on this project and especially for their help as reviewers for the experimental portion of the project.

I also thank Wenbo He for her suggestion for my semester project that gradually changed into my Master's thesis and Terran Lane for his instruction on machine learning that I used for the recommender system algorithm. Thanks also to George Kelbley and Jeff Bowles for their help setting up the software and servers for my experiment.

Finally, I thank Melanie Moses for introducing me to the world of ant colony algorithms. I also thank her for her patience, encouragement, and always helpful and insightful comments. She has been a great guide in my transformation from a biologist who used computers to a biological computer scientist.

Improving peer review with ACORN: ACO algorithm for Reviewer's Network

By

Mark Flynn

B.A., Biology and Psychology, University of Delaware, 1992

Ph.D. Neurobiology, University of Delaware, 2001

M.S. Computer Science, University of New Mexico

Abstract

Peer review, our current system for determining which papers to accept and which to reject by journals and conferences, has limitations that impair the quality of scientific communication. Under the current system, reviewers have only a limited amount of time to devote to evaluating papers and each paper receives an equal amount of attention regardless of how good the paper is. We propose to implement a new system for conference peer review based on ant colony optimization (ACO) algorithms. In our model, each reviewer has a set of ants that goes out and finds articles. The reviewer assesses the paper that the ant brings according to the criteria specified by the conference organizers and the ant deposits pheromone that is proportional to the quality of the review. Each subsequent ant then samples the pheromones and probabilistically selects the next article based on the strength of the pheromones. We used an agent-based

model to determine if an ACO-based paper selection system will direct reviewers' attention to the best articles and if the average quality of papers increases with each round of reviews. We also conducted an experiment in conjunction with the 2011 UNM Computer Science Graduate Student Association conference and compared the results with our simulation. To assess the usefulness of our approach, we compared our algorithm to a greedy algorithm that always takes the best un-reviewed paper and a latent factor analysis recommender-based system. We found that the ACO-based algorithm was better than either of the greedy or recommender algorithms at directing users' attention to the better papers.

Table of Contents

Introduction 1
Methods 4
Simulation 4
Experiment 11
Results 13
Simulation 13
Experiment 24
Discussion 28
Appendix 33
Pseudocode 33
Experiment 33
References 35

Introduction

The peer review system is the cornerstone of the vast majority of modern scientific communication. It is the method for determining which research is suitable for dissemination and where it should appear. Despite the success of the current system, there are some disadvantages of peer review. There are questions of fairness and bias towards established authors, and how big a role chance plays in determining whether a paper is accepted (Neff BD 2006). Computer science publishing is based on conference proceedings. A small group of reviewers is tasked with determining which papers are suitable for presentation at the conference and later inclusion in the proceedings and also assigning them to groups based on the paper's topic. The restricted pool of reviewers means that each reviewer must assess many papers and each paper can only be seen by a few reviewers. Furthermore, each paper receives the same amount of attention from the reviewers regardless of how good the paper is.

Peer review as a principle enjoys tremendous support, but as a process only 8% of those surveyed agreed with how it is actually implemented (Chubin and Hackett 1990). Peer review purports to objectively determine which papers are suitable for publication. However, when this has been tested experimentally, the probability that reviewers agree with each other is no better than chance

(Rothwell and Martyn 2000) and that the process is very poor at identifying flaws (Godlee, Gale et al. 1998).

We propose to implement a new system for computer science peer review based on ant colony optimization (ACO) algorithms. ACO algorithms have been used to efficiently allocate limited resources, such as for the traveling salesperson problem (TSP), engineering applications such as the design of VLSI chips (Arora and Moses 2009), network routing (Kwang Mong and Weng Hong 2003; Marco Dorigo 2006) and data mining (Parpinelli, Lopes et al. 2002). The ACO algorithm is a metaheuristic that has been used to solve combinatorial optimization problems. Each ant explores the solution space and leaves a pheromone trail that indicates the quality of the solution it has found. Other ants encountering these trails will preferentially follow the trails left by the ants who found the best solutions. As more ants find the better trails, those trails become reinforced until the colony as a whole converges on the best solution.

ACO algorithms were inspired by observations of foraging behavior in ants. While individual ants have only a limited view of their surroundings, the colony as a whole can arrive at a globally near-optimal solution. Ants communicate through the exchange of chemical signals called pheromones. Ants lay these volatile compounds on the ground if they find a resource that is useful to their nest, e.g. food, a new nesting site, building materials (Goss, Aron et al. 1989; Beckers, Deneubourg et al. 1992). When other ants encounter a pheromone trail, they preferentially follow it. If the trail leads to a food source, these ants leave a pheromone trail of their own as they return to the nest. In this

way, the best food sources attract ants which leave pheromone trails that attract more ants. This produces a positive feedback effect that allows the ants to quickly exploit the available resources (Marco Dorigo 2006). ACO algorithms work by reducing the size of a search space so that bad solutions become highly improbable (Marco Dorigo 2006). Ant colony optimization algorithms are an efficient strategy for integrating multiple individual decisions into a single good solution.

We used this property of ant colonies to direct the attention of the ants as inspiration for our modification to the current peer review system. In our model, each reviewer has a set of ants that goes out and finds articles. The reviewer assesses the paper that the ant brings according to the criteria specified by the conference organizers and the ant deposits pheromone that is proportional to the quality of the review. Each subsequent ant then samples the pheromones and probabilistically selects the next article based on the strength of the pheromones.

We used an agent-based model (ABM) to determine if an ACO-based paper selection system will direct reviewers' attention to the best articles and if the average quality of papers increases with each round of reviews. ABMs are useful for modeling systems in which complex behaviors emerge from interactions among individual agents with relatively simple behaviors (Grimm, Berger et al. 2006). Additionally, if the goal is to determine which papers will be accepted, the model can also be used to determine which papers exceed a given cutoff. For example, if the conference can only accept the top 40% of the papers they receive; those papers that are closest to the cutoff would receive the most

scrutiny. We also looked at the sensitivity of the model to amount of agreement on paper quality and the degree of trust among the reviewers on convergence of the model on the target paper quality. To assess the utility of our approach, we compared our algorithm to a recommender system based on latent factor analysis. Finally, we compared the ability of all three algorithms to correctly rank the papers according to their pre-determined paper quality.

Methods

Simulation

To test whether our ACO network would be useful for evaluating and sorting papers for a CS conference, we simulated the peer review process using an agent-based model. The agents in this model are ants that search through the papers and bring them to the reviewers. Paper quality was modeled as a normal distribution. The mean of the distribution was a quality score which could be considered the “ground truth”. This would be the paper’s score if it was reviewed by a large number of reviewers such that further reviews would be unlikely to change the score. As shown in Figure 1, each of these means was drawn from an overall normal distribution in order to reflect the diversity of papers a conference is likely to receive, and the amount of diversity was determined by the distribution’s standard deviation. This overall standard deviation controlled how dissimilar the paper qualities were. Each paper had its own standard deviation, for its own distribution which determines how widely the reviews for the

paper can vary. The higher the paper's standard deviation, the less likely different reviewers will agree with each other. This was a parameter of the model that we varied to examine the effect of reviewer agreement on the ability of the network to converge on the target paper score.

Another factor we considered was reviewer bias; each reviewer may have a tendency to rate papers either higher or lower than the mean. We modeled this by skewing the normal distribution from which the paper scores were drawn by a factor that was unique for each reviewer. This skew factor was chosen by randomly selecting a factor from a normal distribution with a 0 mean and a standard deviation which was varied from 0 to 4 to determine the effect of the amount of bias on the algorithm's ability to pick the best papers. Each review consisted of selecting a score from this skewed normal distribution.

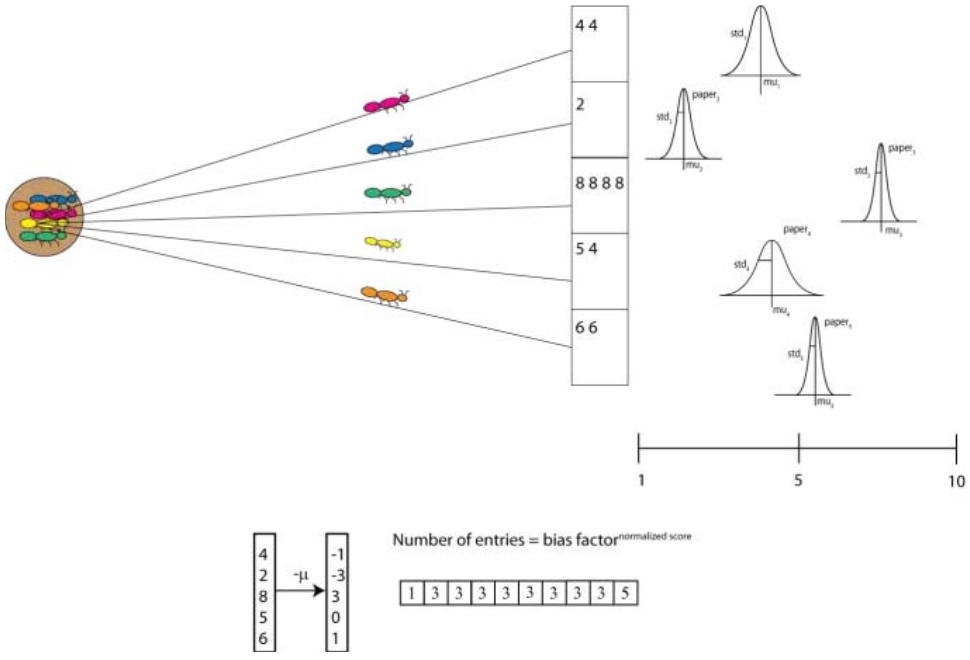


Figure 1. ACO peer review algorithm

To ensure that each paper was reviewed at least once, each ant randomly selected a paper until all papers have received one review. The reviewers selected the score (on a scale of 1 to 10) from the skewed normal distribution and placed a pheromone trail to the paper that is equal to the reviewer's score. For subsequent rounds of reviews, the ants sample the pheromone trails and select the next paper for review probabilistically based on the average quality of the reviews for each paper up to that point. Since paper quality is static, we did not include evaporation as has been done for other implementations of the ACO algorithm (Marco Dorigo 2006). The probability of a $paper_i$ being selected by $reviewer_j$ was determined using an exponential equation:

$$p_{ij} = \begin{cases} \dfrac{b^{(\bar{\tau}_i - \tau_\mu)}}{\sum_{i \in N_j} b^{(\bar{\tau}_i - \tau_\mu)}}, & i \in N_j \\ 0, & otherwise \end{cases} \quad (1)$$

Where p_{ij} is the probability that $paper_i$ is selected by $reviewer_j$, N_j is the list of unread papers for $reviewer_j$, $\bar{\tau}_i$ is the mean pheromone value for the i^{th} paper, τ_μ is the average pheromone value for all papers. The base factor, b, reflects how much the reviewers trust the opinions of the other reviewers. The effect of this equation is that the higher the base factor, the more likely that the better papers are selected. The exponential form was chosen for equation 1 to implicitly model the positive feedback effect of ants reinforcing pheromone trails. The base factor was the same for each paper. We varied b between 1 and 2 to investigate the role of trust in directing reviewers' attention to the best papers.

Parameters	Value
Pheromones	1-10
Base	1-2
Mean of paper standard deviation distribution	0-4
Standard deviation of standard deviation distribution	0.5
Mean of paper distribution	5
Standard deviation of paper distribution	3
Mean of distribution for skew values	0
Standard deviation of distribution for skew values	0-4

Table 1. Ranges and values of parameters used in simulation

To assess how our algorithm matches up to other methods for selecting papers, we compared our algorithm with other methods for selecting papers to be reviewed. First we used a greedy algorithm that always selected the best paper that the reviewer has not reviewed yet. Unlike our ant-based algorithm, papers were selected deterministically based on the quality of the previous reviews, instead of probabilistically. Next, we tried recommender system algorithms, which are used extensively by online retailers to direct customers to products they might like (Melville P and Sindhwani 2010). We used a type of recommender system called collaborative filtering where the previous history of reviews is used to predict whether a user will like other items. We used a type of collaborative filtering called latent factor analysis (Bell R, Koren Y et al. 2009) to detect underlying patterns in the user-response matrix (URM) to predict ratings for un-reviewed papers. Reviewers were directed to the papers that the algorithm predicted would get the highest score for that reviewer. One problem inherent to collaborative filtering is making recommendations when there is no history. We used the greedy algorithm when the recommender algorithm was unable to select a new paper. This is not a problem for the ACO algorithm since it assumes that all users will have similar opinions about all of the items.

To determine the latent factors in the pattern of user-item reviews, we used an algorithm based on singular value decomposition (SVD) (Bell R, Koren Y et al. 2009; Melville P and Sindhwani 2010; Grigorik 2011). The algorithm consisted of taking the SVD of the URM which decomposes it into 3 matrices, U,

S and V^T. U and V are orthogonal rotation matrices while S is a scaling matrix. The first *k* singular values are used to embed each user in *k* space, where *k* < the rank of the matrix. This decomposition can be used to approximate the original URM. This space is a lower-dimensional model that captures some underlying features of the original data that we can use to compare the similarity of users. These features can then be used to predict a user's opinion about an item they have not seen yet.

Once the lower order matrix is computed, each user is compared to all other users using cosine similarity, as shown in equation 2:

$$w_{a,u} = \cos(\vec{r}_a \cdot \vec{r}_u) = \frac{\vec{r}_a \cdot \vec{r}_u}{\|\vec{r}_a\| \times \|\vec{r}_u\|} \quad (2)$$

Where $w_{a,u}$ is the similarity between the active user *a* and *u*, one of the other users being compared to the active user, $\vec{r}_a$ are the ratings by the active user and $\vec{r}_u$ are the ratings of the comparison user. This similarity was used to calculate missing entries in the URM by taking a weighted average of the ratings by users in the active users' similarity neighborhood as show in equation 3:

$$p_{a,i} = \sum_{u \in k} (r_{u,i} - r_{a,i}) \times w_{a,u} \quad (3)$$

Where $p_{a,i}$ is the prediction for the active user for the i^{th} item.

We compared the ability of each algorithm (ACO, greedy and SVD) to pick the best paper given the variability of the papers and the biases of the reviewers. We determined the correlations between the *a priori* paper quality, the "ground truth" that was determined before the simulation was run, and the number of reviews the paper received. We also determined the correlation between paper

quality and the average quality of the papers reviewed for each round of reviews. Each round was considered one time step and the pheromone trails were updated after each review. Each simulation was run 20 times to average the variability due to the stochastic elements of the simulations. We analyzed the response of the three algorithms to varying levels of reviewer bias by varying the standard deviation of the zero-mean normal distribution from which the skew factors were drawn and the standard deviation of the distribution from which the paper means are drawn. The skew factor for each reviewer determines the amount the paper score distribution deviates from the normal distribution, while the standard deviation determines the width of the normal distribution. Each paper would then consist of a unique distribution for each reviewer determined by the mean and standard deviation for that paper and the skew factor for the reviewer.

Another issue we investigated was how many reviews are necessary to rank the papers. Each paper was ranked according to their true value as determined in the initialization procedure. After each review, this true ranking was compared to the how each algorithm actually ranked the papers. The sum of the absolute difference between the two rankings, divided by the number of papers was calculated to determine the average error per paper for each algorithm.

$$r_d = \frac{\sum |q_r - p_r|}{N} \tag{4}$$

Where r_d is the average rank difference, q_r is the rank for each paper according to the value assigned to that paper at the beginning of the simulation,

p_r is the paper ranking according to the algorithm (ACO, greedy or recommender) and N is the total number of papers.

Pseudocode:

Initial Phase:

For i = 1 to number of papers
    Randomly select $reviewer_j$
    Assign $paper_i$ to $reviewer_j$
    Evaluate $paper_i$
    Add $paper_i$ to $reviewer_j$'s list of papers read
    Place pheromone for $paper_i$.
    Remove $reviewer_j$ from list of initial reviewers
End

ACO phase:
For j = 1 to number of reviewers
    For k = 1 to number of ants
        Remove papers that $reviewer_j$ has already seen
        Choose $paper_i$ to read with probability p_{ij} given by equation 1
        Evaluate $paper_i$
        Add $paper_i$ to $reviewer_j$'s list of papers read
        Place pheromone for $paper_i$.
    End
End

Experiment

We tested the ability of our algorithm to pick the best paper by conducting an experiment in conjunction with the 2011 UNM Computer Science department graduate student symposium. A call was sent to all graduate students and faculty to review papers and a total of 7 reviewers agreed to review the 10 papers from the conference. Each reviewer was asked to review 2-3 papers and to give each paper a score between 1 and 10. The reviewers entered their

reviews into a web form written in PHP and all of the information for the experiment was stored in a MySQL database (see Appendix for pseudocode). Matlab was used to implement the ACO algorithm. Just as for the simulations, for the first round each paper was randomly assigned to reviewers until all papers received one review. After the initial round, the ACO algorithm was used to select the next paper for the reviewer to evaluate. Also as for the simulation, the score given by the reviewer was used for the pheromone values. The exponential bias equation (eqn. 1) was used for paper selection with the base of the exponent set to 2, which simulations showed to effectively direct reviewers to the best rated papers.

After each review was entered into the web form, the PHP script wrote the reviews to the MySQL database and then wrote all of the reviews from the database to a text file. Matlab read in the reviews and executed the ACO algorithm. The ACO algorithm selected the next paper for the reviewer to evaluate and the PHP script displayed a link to this paper on the server where the papers were stored. The Matlab program wrote the list of papers each reviewer had read and the list of pheromones for each paper to a text file. After the next review, this list of papers and pheromones was read into Matlab for the next review.

Results

Simulation

Our goal was to determine whether an ACO algorithm could direct reviewers' attention to the simulated papers that were deemed most important. First, we tested the ability of our algorithm to determine which papers were the best (based on the quality assigned at the beginning of the simulation) and how sensitive this determination was to variation in the parameters. The base factor (*b*) was set to 2 and the standard deviation of the paper reviews was set to 2.

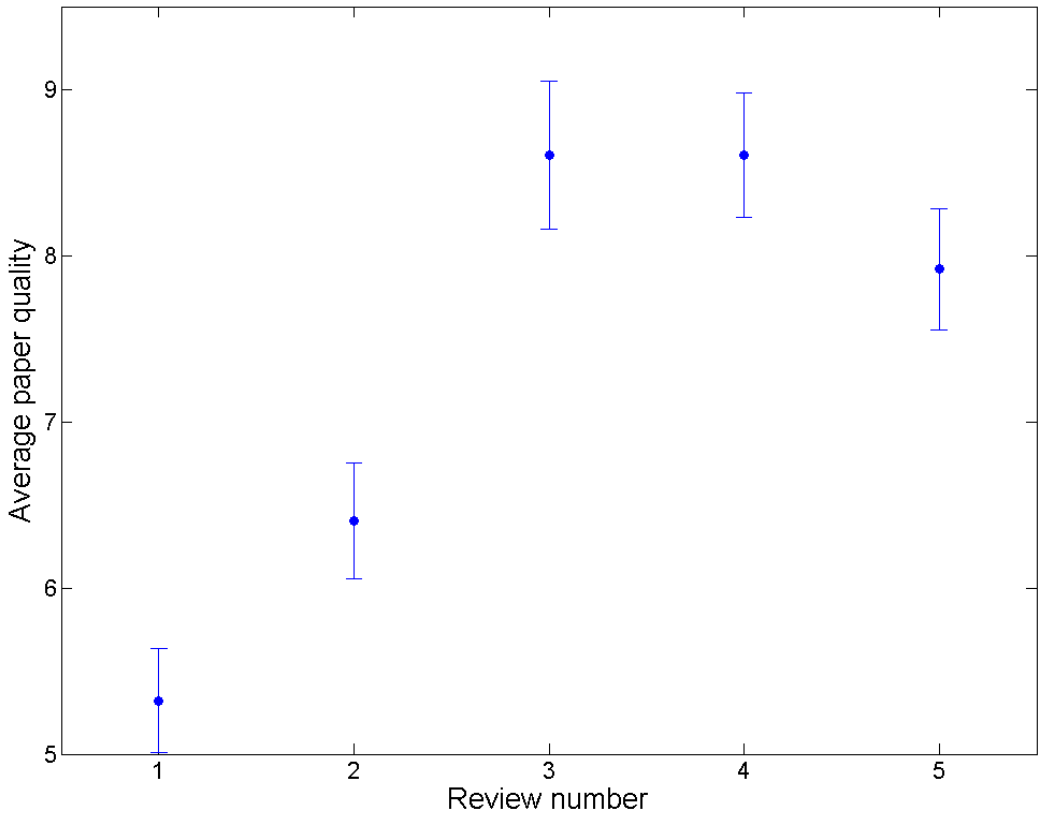


Figure 2. Plot of average paper quality (±SEM) for each round of reviews. Average quality improves with time

We found that there was a positive correlation between how many papers a reviewer has evaluated and the quality of the papers the reviewer receives. As seen in Figure 2, after each round of reviews the better papers were more likely to be selected for review. However, the increase in quality plateaued after the third review. This was because each reviewer could only evaluate a paper once and so once they have seen all of the best papers there could be no further increase in quality. The effects of changing the number of reviewers or papers were not investigated to determine how they affect the plateau in paper scores. We also found a positive correlation between the quality of papers and the number of reviews the paper received. Figure 3 shows that the number of ants that visited a paper depended on the pre-determined ground-truth quality of the paper.

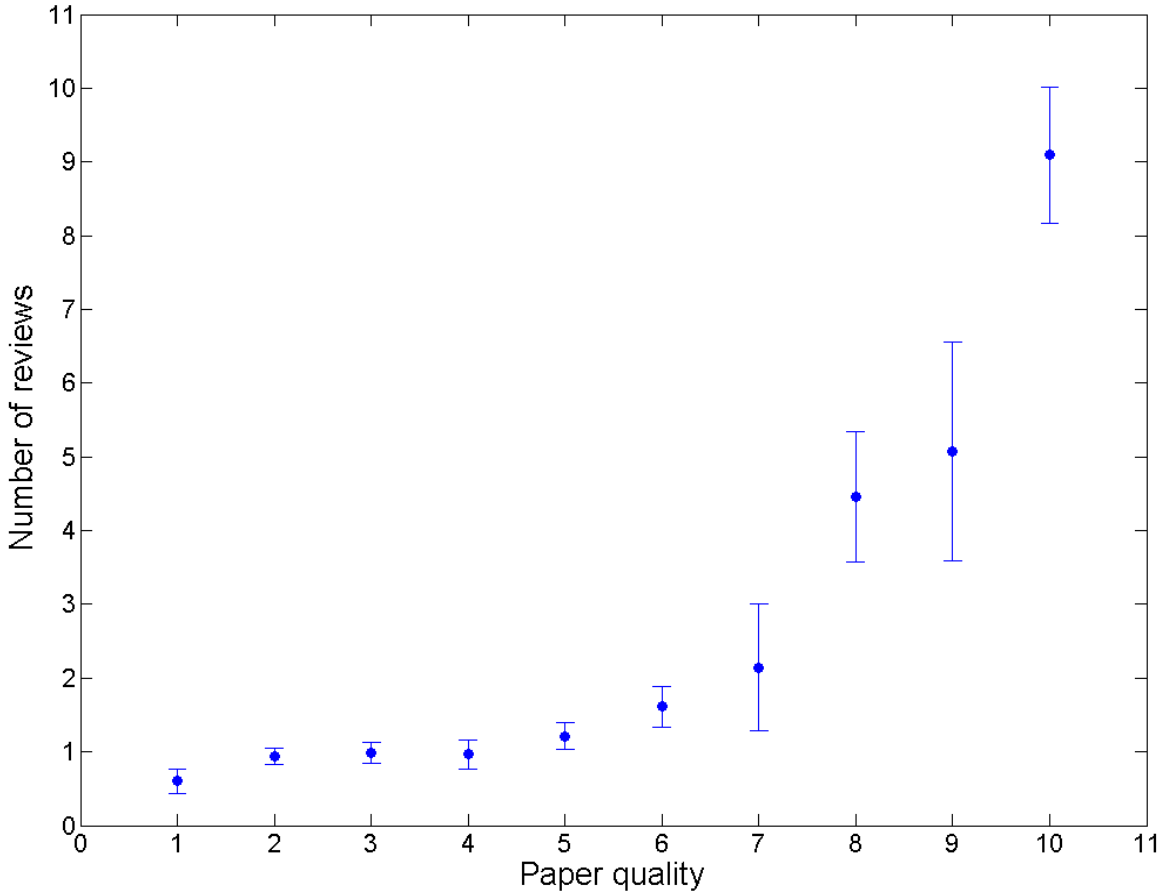


Figure 3. Plot of number of reviews (±SEM) for each paper of a given quality. Higher quality papers receive more reviews

Two important factors that affect the ability of the model to select the best papers are 1) the amount of agreement the reviewers have on the quality of the papers and 2) how much they need to rely on each other's judgment for the algorithm to work. Too much disagreement could mean that the reviewers would not see an improvement in the paper quality, or if the goal is to decide which papers to accept, convergence on the target paper quality. We modeled this by varying the standard deviation for the normal distribution that paper scores are drawn from. This interacts with the degree of trust between the reviewers. While

too little trust could mean that reviewers would receive papers that were selected at random instead of by how much other reviewers liked the papers, too much trust could amplify the effects of disagreement among reviewers. To explore these interactions, we ran all combinations of base factors between 1 and 2, in increments of 0.25 (completely random and completely deterministic paper selection) and paper standard deviations between 0 and 4 in increments of 0.1 (complete agreement and complete disagreement) and determined the correlations between the paper quality and the number of reviews a paper received and between paper quality and the improvement of average paper quality over time. 20 simulations were run with each combination to average out the stochastic elements of the paper selection process.

To determine the interaction between base factor and the standard deviation of the paper scores, the Pearson product-moment correlation coefficient was computed for each combination of base factor and standard deviation. Figures 4 shows the effects of these interactions on the correlation of the number of papers a reviewer has evaluated up to that point and the quality of the papers reviewed. There was a large increase in this correlation between 1 and 1.25 and then the correlation plateaued above 1.5. This correlation was less sensitive to decreasing the amount of agreement among reviewers. While there were significant correlations between the standard deviations of the paper distributions and increase in paper quality over time, this effect decreased as *b*

increased (Table 2).

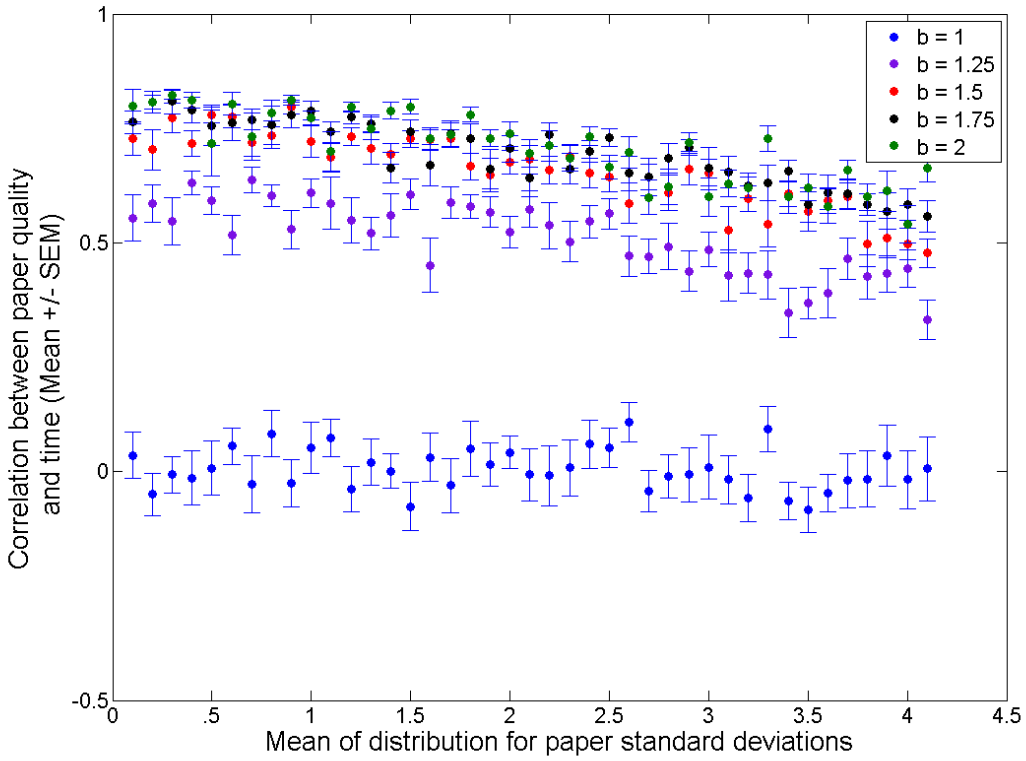


Figure 4. Decreasing the influence of other reviewers' opinions on paper selection decreases correlation between paper quality and the increase in paper quality over time

	b = 1	b = 1.25	b = 1.5	b = 1.75	b = 2
R	-0.1655	-0.8185	-0.9007	-0.9106	-0.8550
P	0.3010	0.0000	0.0000	0.0000	0.0000
Percent change	-84	-40	-34	-27	-18

Table 2. Percent change, correlations (R) and significance values (P) for paper quality vs. time

Figure 5 shows the corresponding effects on the total number of reviews a paper receives. The number of reviews a paper received was also very dependent on the *b* for values between 1 and 1.25 and plateaued above 1.5. However, the effect of increasing the variability of the reviews was much less; there were only significant decreases for values of *b* equal to 1.5 or below (Table 3).

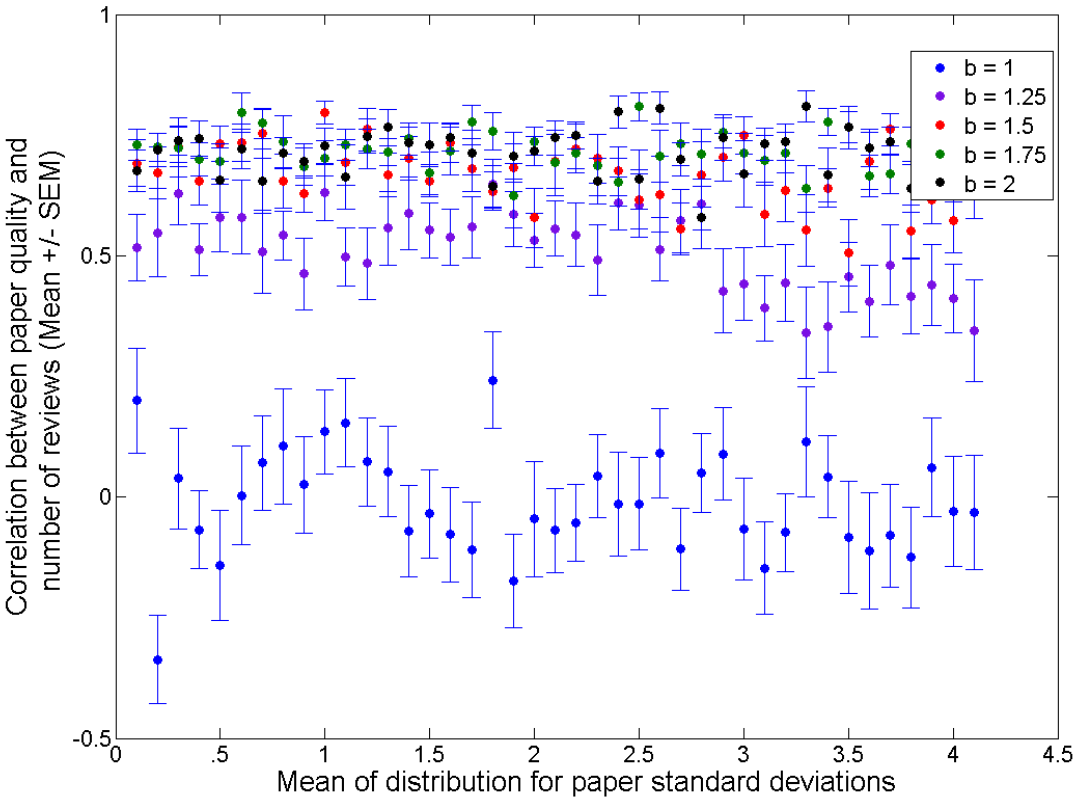


Figure 5. Decreasing the influence of other reviewers' opinions on paper selection decreases correlation between paper quality and the number of reviews a paper receives.

	b = 1	b = 1.25	b = 1.5	b = 1.75	b = 2
R	-0.1313	-0.6248	-0.4827	-0.1910	0.0431
P	0.4131	0.0000	0.0014	0.2316	0.7891
Percent change	-116	-33	-8	-2	2

Table 3. Percent change, correlations (R) and significance values (P) for paper quality vs. number of papers

We compared the performance of the ACO, greedy and recommender algorithms in their ability to find the best paper. Figure 6 shows that all three were able to direct the ants to the best papers, with the ACO and greedy algorithms outperforming the recommender system.

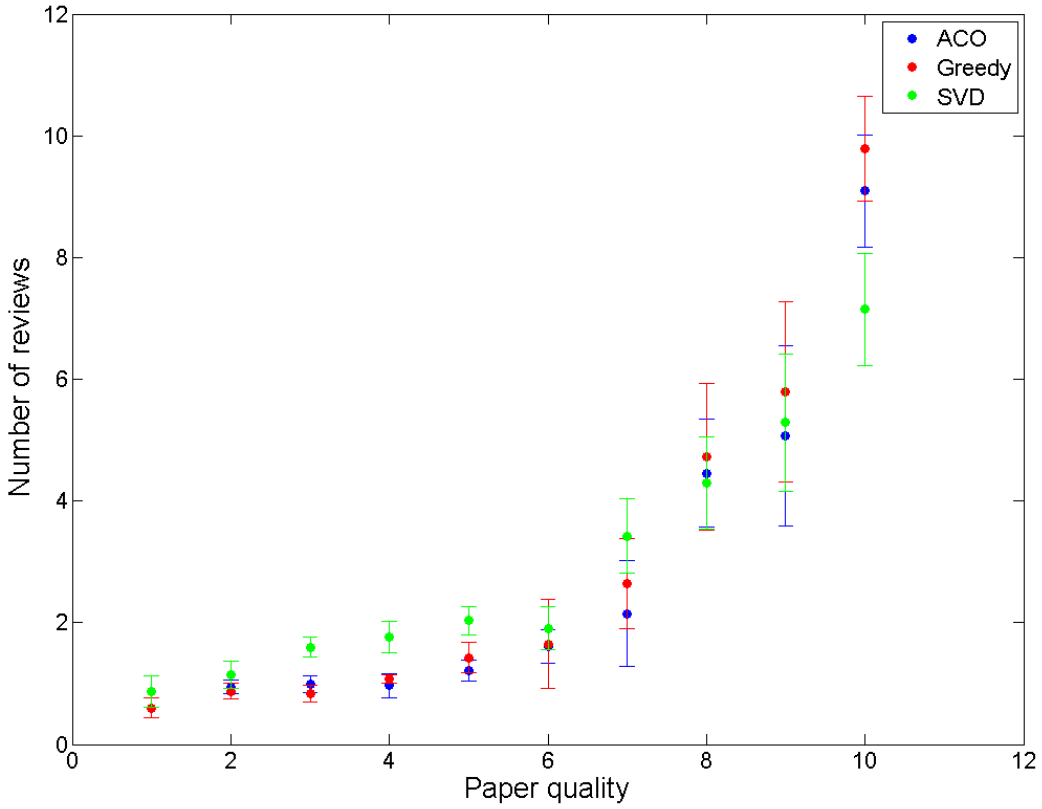


Figure 6. Comparison of ACO, greedy and recommender algorithms

To more accurately model the individuality of the reviewers, we incorporated a skew factor to the normal distribution from which the paper scores were drawn. The skew factors for each reviewer were drawn from a normal distribution so that most reviewers drew their scores from a minimally biased distribution while a few drew their scores from distributions whose modes were

well above or below the mean of the distribution. The standard deviation of this distribution was varied from 0 to 4 to investigate the effect of increasing the bias on the ability of the algorithm to find the best paper and for the average paper quality to increase over time. We used this procedure to compare the three different algorithms, ACO, greedy and recommender. None of the algorithms were affected by the amount of skew in the range that was tested ($p = 0.6344$, 0.1632, 0.3079, respectively). Both the ACO and greedy algorithms were significantly different from the recommender algorithm ($p < 0.0001$ for both) while the ACO algorithm was better than the greedy algorithm ($p < 0.001$).

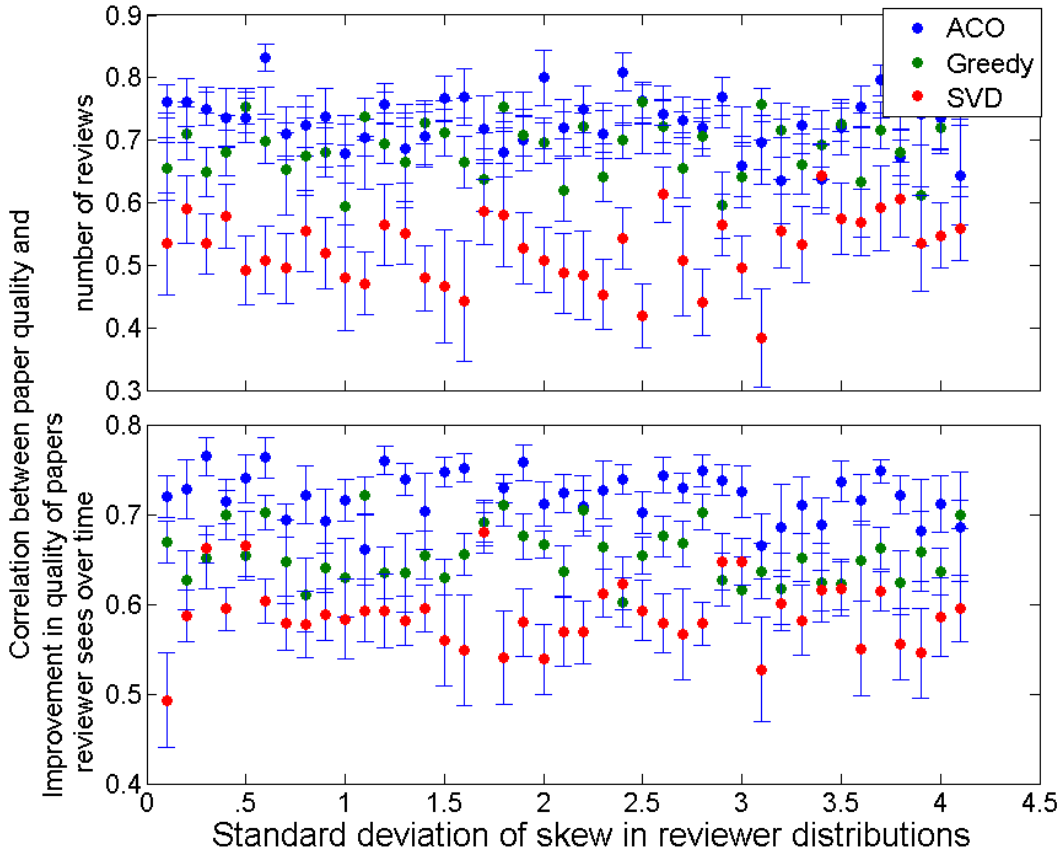


Figure 7. Increasing the skew of the distribution of the reviews does not affect performance of ACO algorithm.

We tested the abilities of all three algorithms to correctly rank the papers by quality. We compared the assigned paper values to how each algorithm ranked them and found that, while all three algorithms were able to rank the papers fairly well, the ACO algorithm made fewer errors than the greedy and recommender algorithms (Figure 7).

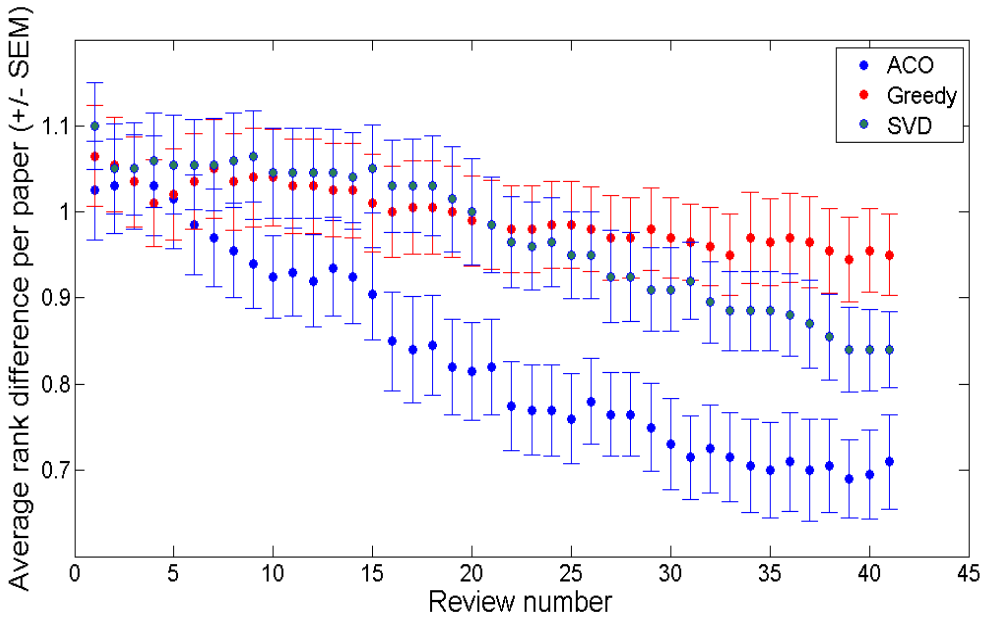


Figure 8. Difference in paper rankings for ACO, greedy and recommender algorithms from their true rankings.

Experiment

Table 4 shows the 18 reviews from the 7 reviewers from our experiment with the UNM Graduate Student Symposium. Numbers indicate the overall quality of the paper where '0' indicates no review for that paper by that reviewer. The mean score was 7.6, the standard deviation was 1.12 and the range was between 6 and 10. The number of papers each reviewer evaluated and the number of reviews each paper received varied between 1 and 3. Three papers received only 1 review, 3 received 2 reviews and 4 received 3 reviews. Unfortunately, the number of reviews was very small. Also, the initial round of reviews included 12 papers due to the fact that some papers were assigned more than once during the initial round because of the variability in the length of time it took reviewers to record their evaluations. The mean score of the reviews

after the initial round, when the ACO algorithm determined which paper for the reviewer to evaluate next, was 7.1667. The reviews after the initial round were not statistically different from the reviews during the initial round ($p = 0.128$).

Paper number

Reviewer	1	2	3	4	5	6	7	8	9	10
A	0	6	9	0	0	0	0	0	7	0
B	0	0	10	0	8	9	0	0	0	0
C	0	0	0	7	0	0	8	7	0	0
D	7	0	0	0	0	0	0	0	7	7
E	8	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	6	8
G	8	0	0	6	0	0	0	9	0	0

Table 4. Reviews from experiment

Paper	Mean score	Score standard deviation
1	7.7	0.58
2	6	0
3	9.5	0.71
4	6.5	0.71
5	8	0
6	9	0
7	8	0
8	8	1.4
9	6.7	0.58
10	7.5	0.71

Table 5. Score summary

Discussion

We found that our system for changing the peer review process can successfully direct reviewers' attention to the best papers. The best papers received the most reviews, there was an increase in paper quality over time and the algorithm could rank the papers correctly. We found that the ACO algorithm was robust to decreasing reviewer agreement while remaining sensitive to changes in the base factor necessary for emphasizing the best papers. Changing the goal to selecting papers nearest to a cutoff for acceptance is equivalent to selecting the best paper, since it is only a matter of rescaling the paper scores by distance from the cutoff. This would favor the papers nearest the cutoff instead of the papers that received the best reviews. Our experiment with the UNM graduate student symposium, however, did not have sufficient time to adequately test whether the ACO algorithm is useful for an actual computer science conference.

While all three algorithms were able to find the best papers, the ACO-based algorithm was more robust to inaccurate reviews. The probablistic nature of the ACO algorithm was less sensitive to errors. While both algorithms use positive feedback to increase the likelihood that the best papers are selected for reviews (Dorigo, Maniezzo et al. 1996), the greedy algorithm is more susceptible to errors since it always takes the best paper available and if a low quality paper is given an overly generous first review the network is less able to recover. The probabilistic ACO algorithm exhibits better fault-tolerance because in a sense it is

exploring multiple solutions in parallel (Dorigo, Maniezzo et al. 1996), while the greedy algorithm converges on the best solution at the time, even if it is not close to the optimal. While the reason for the poor performance by the recommender system has not been determined, one possibility could be a problem inherent to this type of algorithm. Collaboration-based recommender systems rely on patterns in the user-item matrix. So, when only a few items have been rated, there is very little for the algorithm to work with. Especially in the beginning, the user-response matrix is very sparse and so it is unlikely that the performance of the recommnder system could be improved.

The concept of pheromones encompasses all criteria important for the reviewing process. This can be extended to include other criteria such as novelty, technique, style and topic. Big heterogeneous conferences with a large diversity of topics could use pheromones to direct reviewers to papers that are more relevant to their expertise. Reviewers could enter information regarding their research interests and then rate papers according to topic attributes along with the quality of the paper. The pheromones would then be a vector incorporating all of the criteria relevant to the conference.

One can make the case that time is the most precious resource we have. The duties and responsibilities of scientists are never-ending. Any steps taken to conserve this resource will benefit authors, reviewers and editors. Authors will benefit from more thoughtful and informative reviews. Automating the work involved in deciding which reviewer should review which paper and determining which paper should be accepted will decrease the amount of work for the editors.

Improving the peer review process will increase reviewer participation and enthusiasm. When reviewers feel that their hard work contributes to the overall quality of the conference they will be more willing to devote the time and energy necessary to give good reviews. If one accepts the premise that there is an intrinsic quality score for each paper, then the more time a reviewer spends evaluating an article will result in a review closer to this score and so decrease the variance and decrease the number of reviews necessary for each paper.

The experience of conducting our experiment with the UNM graduate student colloquium has resulted in many lessons learned. Reviewers entered their reviews into a PHP-powered web form; this information was stored in a MySQL database as well as text files and a Matlab program actually executed the ACO algorithm. The program would read in the text files, execute the algorithm and write the output and all the variables necessary to keep track of the state of the algorithm. All communication with the MySQL database occurred though PHP. Unlike with the simulation, the papers with the highest scores did not receive more reviews than papers with lower scores. The problem arose from using text files to store the state of the ACO program in between reviewers. Several errors occurred where Matlab was unable to write its output to the text files which resulted in an inappropriate state when it processed the next reviewer. The algorithm, therefore, did not have accurate input when it picked the next papers for reviewers. In the future, it would be preferable for Matlab and MySQL to communicate directly instead of exchanging information via text files and PHP.

The experiment accomplished two things: it provided data about the distribution of reviews and it allowed us the opportunity to construct a framework for conducting peer review experiments. In the future, we would like to use this framework to test the conclusion of the simulation that the ACO algorithm is better at selecting papers than the greedy algorithm.

The goal of this project was to demonstrate the utility of our ACO algorithm in directing reviewers' attention for conferences where a pool of reviewers assesses the quality of a group of submitted papers. However, this approach would likely be useful for other peer review processes. The algorithm could be adapted for review of articles for biomedical journals. Currently, an editor chooses 2 reviewers for each paper, and if they disagree on whether the paper should be accepted or not, either the editor makes a decision based on these two reviews or the paper is sent to a third reviewer. However, this process assumes that the reviewers will agree on what should be published which might not be a good assumption. Rothwell and Martyn (2000) found that agreement between reviewers for two clinical neuroscience journals were no better than chance (Rothwell and Martyn 2000). In addition, it has been found that in a study where errors were intentionally introduced into papers, that on average only 25% of the errors were found (Godlee, Gale et al. 1998).

These problems could be mitigated by expanding the number of reviewers for each paper and using the ACO algorithm to handle the extra work involved in integrating the reviews. Instead of picking the best two people to review an article, let a larger number of somewhat knowledgeable people become

reviewers. The problem of picking two experts is becoming more difficult as research becomes more interdisciplinary. A larger group of reviewers with a variety of research experiences and perspectives would be better at evaluating articles that cannot be neatly categorized. While this has never been tested in peer review, it has been shown for other highly technical tasks such as interpreting lung x-rays that a large number of movie opinions were superior to a small number of experts (Surowiecki 2004). The ACO algorithm could assign papers to reviewers and rank each article for the editor.

Some things we would like to improve for the future would be explicitly modeling the reviewers. One example would be if reviewers change their behavior over time in response to an expected increase in paper quality. Another example would be to determine the effects of groupthink by creating correlations between reviewers. Currently, each review is completely independent of other reviewers' opinions. What would happen if there was communication between reviewers about papers? Poor reviews (either well above or well below the mean) do not affect other reviews; they are just as likely to be below as above the mean. It is possible that correlations in errors could distort how well the papers are ranked.

Another possibility we would like to investigate is using the ACO algorithm as a method for decreasing the number of reviews the reviewers need to perform. As shown in Figure 7, the quality of the paper ranking by the ACO algorithm does not improve after the 30th review and so further reviews do not add any additional information. There are two possibilities we will investigate,

one is to use the variance of the scores in the selection process. The overall evaluation of papers that receive very similar scores are unlikely to change with further reviews and will be removed so that the reviewer's attention can focus on the papers whose fate is more uncertain. Another is to remove papers whose rank remains the same after multiple reviews.

We would also like to include some of the best practices from the peer review process among the journals. For example, the Artificial Intelligence conference includes the reviewer's assessment of their experience with the topic of the paper, so that the views of experts and those not as familiar with a particular topic can have their evaluations properly weighted. Another practice we will investigate is the use of multiple levels of reviews, where the results of the review are passed to a higher level for further review.

Appendix

Pseudocode

Experiment

Initial Phase:

- Register reviewer
    - Add username to MySQL database
- Reviewer logs in
- PHP queries MySQL to check if all papers have received at least one review
    - If so, enters ACO phase
    - Else, continues Initial Phase
- A paper is selected randomly from the list of papers that have not been read

- Reviewer enters review into PHP web form
- PHP enters review into database
    - Updates list of papers reviewer has evaluated
    - Updates list of pheromones for paper
- PHP writes list of papers reviewed and list of pheromones to text files

ACO phase:

- Reviewer logs in
- Reviewer enters review into PHP web form
- PHP enters review into database
    - Updates list of papers reviewer has evaluated
    - Updates list of pheromones for paper
- PHP writes list of papers reviewed and list of pheromones to text files
- PHP starts Matlab program that executes ACO algorithm
    - Read in text files with
        - list of papers reviewed
        - list of pheromones
    - Remove papers that reviewer has already read
    - Use eq. 1 to select next paper
    - Write to text file
        - list of papers reviewed
        - list of pheromones
- Web page is generated with link to next paper for reviewer to read

References

Arora, T. and M. Moses (2009). Using ant colony optimization for routing in VLSI. 1st International Conference on Bio-Inspired Computational Methods Used for Difficult Problem Solving: Development of Intelligent and Complex Systems. AIP Conference Proceedings.

Beckers, R., J. L. Deneubourg, et al. (1992). "Trails and U-turns in the Selection of a Path by the Ant Lasius niger." J. theor. Biol 159(4): 397-415.

Bell R, Koren Y, et al. (2009). "Matrix Factorization Techniques for Recommender Systems." IEEE Computer Society 42(8): 30-37.

Chubin, D. and E. Hackett (1990). Peer review and the printed word. Peerless Science: Peer Review and U.S. Science Policy. Albany, NY, SUNY Press.

Dorigo, M., V. Maniezzo, et al. (1996). "The Ant System: Optimization by a colony of cooperating agents." IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics 26(1): 29-41.

Godlee, F., C. R. Gale, et al. (1998). "Effect on the quality of peer review of blinding reviewers and asking them to sign their reports: a randomized controlled trial." JAMA 280(3): 237-240.

Goss, S., S. Aron, et al. (1989). "Self-organized shortcuts in the Argentine ant." Naturwissenschaften 76(12): 579-581.

Grigorik, I. (2011). "SVD Recommendation System in Ruby." from http://www.igvita.com/2007/01/15/svd-recommendation-system-in-ruby/.

Grimm, V., U. Berger, et al. (2006). "A standard protocol for describing individual-based and agent-based models." Ecological Modelling 198(1-2): 115-126.

Kwang Mong, S. and S. Weng Hong (2003). "Ant colony optimization for routing and load-balancing: survey and new directions." Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 33(5): 560-572.

Marco Dorigo, M. B., and Thomas St¨utzle (2006). "Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique." IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE 1(4): 39.

Melville P and V. Sindhwani (2010). Recommender Systems. Encyclopedia of Machine Learning. G. Sammut and G. Webb. Berlin, Springer-Verlag.

Neff BD, O. J. (2006). "Is Peer Review a Game of Chance?" Bioscience 56(4): 333-340.

Parpinelli, R. S., H. S. Lopes, et al. (2002). "Data mining with an ant colony optimization algorithm." Evolutionary Computation, IEEE Transactions on 6(4): 321-332.

Rothwell, P. M. and C. N. Martyn (2000). "Reproducibility of peer review in clinical neuroscience. Is agreement between reviewers any greater than would be expected by chance alone?" Brain 123 (Pt 9): 1964-1969.

Surowiecki, J. (2004). The Wisdom of the Crowds. New York, NY, Anchor Books.