Brainstorming, Negotiating and Learning in Group Decision Support Systems: An Evolutionary Approach

Jackie Rees
Purdue University
Krannert Graduate School of Management
West Lafayette, IN 47907
jrees@mgmt.purdue.edu

Gary Koehler
University of Florida
Decision and Information Sciences
Warrington College of Business
Gainesville, FL 32611
koehler@ufl.edu

Abstract

Certain tasks undertaken by groups using Group Decision Support Systems (GDSS) can be viewed as search problems. These tasks involve arriving at a solution or decision where the problem is complex enough to warrant the use of computerized decision support tools. Also, the task or situation must require more than one person to adequately address the problem. For these types of GDSS tasks, we propose to model the brainstorming, negotiating and learning processes undertaken by the group as a simple genetic algorithm.

The simple genetic algorithm is a generalized search technique that is based on the principles of evolution and natural selection. Simply put, the best points in the search space are more likely to be selected and combined through genetic operators to determine new points. We propose that groups using GDSS to address certain tasks behave like a simple genetic algorithm in the manner in which possible solutions are generated, enhanced and altered in attempting to reach a decision or consensus.

I. Introduction

Groups of individuals meeting to solve particular problems can be viewed as searching for a solution within some sort of solution space [6]. This search space is most likely highly complex, otherwise a group would most likely not be used. Group support systems, in particular Group Decision Support Systems (GDSS) have been used to assist groups in their problem-solving efforts. Certain tasks require the group to decide upon one outcome or course of action. For these tasks, we can model the group activities of brainstorming, learning and negotiating as a simple genetic algorithm.

Hirokawa and Johnson proposed that the group decision-making process is itself an evolutionary process [8]. Hence, the idea that groups undergo change and that the initial ideas or proposals submitted during a brainstorming session are subject to adaptation is not a new idea but has not been formally incorporated into analytical models for GDSS.

This paper will describe an analytical model for groups using GDSS using a simple genetic algorithm (GA) as the basis of the model. We test this model using experimental data and will present the results of tests of hypotheses linking GA parameters to one specific set of GDSS variables.

A. Background

A brief overview of several analytical GDSS models is presented. The relative strengths and weaknesses of the models are highlighted along with the justification for why another analytical model would be useful for both researchers and practitioners. We present a brief background on GAs as they are used in this particular study. The mechanics, underlying theory as pertaining to this research and justification for why they should be used as a modeling tool are discussed.

B. Analytical Models for GDSS

Group decision support systems are designed to support group decision-making through specialized software, hardware and decision support tools. DeSanctis and Gallupe [3] defined GDSS as a combination of computer, communications and decision technologies working in tandem to provide support for problem identification, formulation and solution generation during group meetings. For our purposes, we do not restrict the term GDSS to the traditional decision room and facilitator. Meetings can be either Face-to-Face (proximate) or Computer-Mediated (distributed). The meeting time is assumed to be same-time (synchronous), although extensions could easily be made to different-time (asynchronous) meetings.

Valacich and Dennis [16] presented a simple mathematical model of electronic brainstorming using GDSS. Their model presents GDSS brainstorming as the ideas generated by a group individuals, each working alone, accounting for process losses and process gains[1]. In other words, "..group performance is a function of individual performance minus process losses plus process gains," ([16] pp.64). Their model was one of the earliest models to provide analytical insight into a particular GDSS process. The shortcoming of the model is that it can provide no insight into the expected behavior of the system, where the system is composed of the group members, the environment, the task under consideration, the reward (internal or external) tied to decision quality and the final decision itself.

Perhaps the most closely related GDSS research to this particular research project is the economic analysis of GDSS [6]. This work was preceded by research on distributed GDSS by Gavish et al. [4],[5] where brainstorming and other GDSS activities were closely examined. One of the important features of this model is that it considers GDSS use by groups to be in the format of a search problem with a very large search space. According to their model, every feasible solution has a payoff, which must be balanced with the cost of performing the search. Another aspect of interest in their model is the discussion of a "trigger phenomenon" [6]. This is the case when an original idea "triggers" a new line of reasoning or discussion. The model also addressed the probability of finding a solution, the expected net benefit of finding a particular solution, stopping criteria, and the marginal value of group size [6]. One drawback to the economic model is that the model fails to describe how the proposed solutions and ideas change over time. By capturing this adaptation, we can better manage group decision-making processes.

[1] See [14] for the detailed process model for GDSS

C. Genetic Algorithms

Genetic algorithms (GAs) are general-purpose search algorithms driven by the basic principles of Darwinian natural selection and evolution. Search is performed from a population of agents, rather than the traditional single point. Such agents, called strings, points or chromosomes, explore a space using three basic operations. First, strings are evaluated according to a given objective function. This evaluation, or fitness, influences the likelihood of the proportion of the string in the next time series, or generation. Fitter strings generally have a greater chance of being stochastically selected for the next generation. Second, selected strings are recombined, or crossed, in hopes of discovering better or fitter strings by combining genetic material. Third, the selected strings are randomly mutated to replace any lost diversity after selection and crossover. As such, GAs are a stochastic search technique.

Selection occurs similar to that of asexual reproduction in the natural world. Chromosomes that are deemed "fit" by measure of a pre-defined fitness function are stochastically more likely to be represented in future populations. Strings are thus drawn with replacement from the current generation with bias according to a fitness measure and placed into the next generation. This method is known as stochastic sampling with replacement, or more commonly referred to as "roulette wheel" selection. Other commonly used selection schemes include tournament selection and rank selection. In tournament selection, strings are drawn from the population using the method above in pairs and the string with the higher fitness value is placed in the new population. Rank selection starts by sorting the population according to fitness value. Each string receives new copies that are placed in the new population according to a function of this ordering. Several other variations of selection are discussed in [7].

Crossover implements a mating strategy for the combination of "good" genetic material between fit parents. After the selection procedure is complete, crossover is applied with a predetermined, fixed probability, called the crossover rate. Two members of the new population are paired up, each selected with the probability given by the crossover rate. In the most common crossover scheme, one-point crossover, a single site is uniformly selected with probability $\frac{1}{\ell - 1}$ where ℓ is the length of the string. The parent sub-strings are exchanged on the right-hand side of the crossover site. Two-point crossover works in a manner similar to that of one-point crossover. However, the string is viewed as a ring and two crossover sites are randomly and uniformly selected. The sub-strings demarcated by the two points are thus exchanged. Uniform crossover works slightly differently. Each string is selected for crossover just as in single-point crossover, but rather than selecting a crossover site, each bit in the string is exchanged with the corresponding bit in the other string with probability $2^{-\ell}$ [17]. Other crossover schemes exist and are discussed in [7].

Mutation is the last operation on the population before the next generation is completely formed. In the binary case, mutation simply requires the mutated bit becomes its complement, i.e., 0 becomes 1 and vice versa. Under uniform mutation, mutation is applied with a fixed, pre-determined probability to each gene (each bit) in every string. The mutation rate is usually kept very low in order to keep the search from diversifying too rapidly. Other mutation schemes are available and discussed in [7].

Crossover and mutation play important roles in the search process. The crossover and mutation operators working together direct the search towards promising areas of the search space while avoiding local optima. Crossover acts as a "focusing" operator by combining elements of strings determined to be "more fit" by the selection operation. The idea is to combine two strings that are relatively good solutions and create new strings that contain elements from both desirable parent strings. Mutation, on the other hand, acts to introduce new strings to the search and also recover strings previously discarded. The function of the mutation operator in a mathematical sense is to direct the search away from local optima.

Nix and Vose [13] developed a Markov chain (MC) model for genetic algorithms. Each state of the Markov chain represents a population of the GA. This model provides an exact representation for the expected populations of a GA over time.

II. An Evolutionary Model for Group Decision Support Systems

As mentioned previously, little has been done to incorporate the adaptation of potential solutions into an analytical model for GDSS. As the GA described above is an evolutionary computing technique, we can use the mechanics and the mathematical theory[2] behind the genetic algorithm to better describe the brainstorming, negotiation and learning processes that occur during GDSS use on particular tasks.

Abstractly, a very simplistic GDSS session could involve the following. Given a task (for example, to determine a solution for resource allocation for the organization involving different and potentially conflicting constraints, costs and benefits to specific group members and the departments they represent) ideas or possible solutions are proposed. The better parts of ideas, according to the group, are exchanged, often (but not always) resulting in "even better" ideas. These ideas are again refined and exchanged until the group agrees upon a solution (or agrees to meet further). Occasionally, a unique or quite different solution or idea is proposed (the so-called idea from "left-field"), or the trigger phenomenon discussed earlier. We can describe the idea proposal or generation process similar to selection. The exchange process resembles the crossover process in the GA. Finally, the randomly appearing idea or solution (that varies from the current "line of discussion") resembles a mutation. Obviously the above analogy is simplistic and doesn't incorporate all of the nuances of various GDSS features, but illustrates the most basic GDSS processes.

We propose that group problem solving, when supported by GDSS, can be modeled by a simple genetic algorithm, utilizing selection, crossover and mutation. Each group is represented by a population of strings and each string in the population at time step t represents the solution proposed by group members at time step t. Selection, crossover and mutation operate as above. The solutions proposed a group member can be encoded as fixed-length strings. As the generations evolve, the genetic algorithm tends to find better and better solutions. We

[2] The theory behind the simple genetic algorithm is presented in [17].

believe that GDSS supported groups behave similarly.

We have chosen a simple genetic algorithm as the basis of this model for several reasons. First, GAs are adaptive, meaning there is change over time, in response to the environment, including the fitness function and other constraints. The second reason for using a GA is a body of formal, mathematical theory has been developed to describe the expected behavior of the simple genetic algorithm. If groups using GDSS for particular tasks can be modeled as a GA, this theory could provide numerous insights into the group decision-making process. Variables and different environmental pressures thought to influence the process could be related to GA parameters and then aspects such as the expected behavior of the system could be determined or optimized.

An additional reason for using GAs to model GDSS process is to provide the basis for a computational model that has both stochastic and deterministic properties. Eventually this model could be used to develop simulation studies. These simulation studies can then be used to examine various combinations of GDSS variables prior to laboratory and field experiments, perhaps identifying previously unknown variables or shedding new light on variables previously studied.

A. Research Questions

The underlying hypothesis for this work is that groups using GDSS act like simple genetic algorithms using selection, crossover and mutation. Note in the following that χ stands for "crossover rate" and μ stands for "mutation rate." Our main hypothesis is:

H1: Groups using GDSS act like a simple genetic algorithm.

We will implement this GA using roulette wheel selection, single point crossover and uniform mutation as described above. This hypothesis will be tested using the methodology described in Section D below.

One of the issues facing group work is the social and political forces that affect members of the group, possibly affecting the quality of the group's decision-making. We hypothesize that Face-to-Face (FTF) groups respond more to group or "societal" pressures and will tend to conform. This would lead to similar thought processes being explored in depth, rather than many different (and possibly conflicting) ideas being presented for consideration. Therefore, FTF groups will be more focused. Computer Mediated Communication (CMC) groups are exposed to fewer visual cues meaning a greater sense of anonymity, which could lead to the proposal of possibly very different solutions [14]. Research performed on distributed groups versus proximate groups has found that distributed groups exhibit greater degrees of depersonalization and impulsiveness, lowered inhibition, and generate "…more extreme opinions," ([15] pp. 328), [16]. In other words, CMC groups can be considered more explorative of the solution space than FTF groups. Under these assumptions of social and political motivations, FTF groups would be less likely to present completely new solutions than CMC groups. CMC group members would be more likely to explore alternate but possibly unpopular or politically less favorable solutions.

H2a: FTF groups are likely to propose less explorative solutions than CMC groups.

We will test this hypothesis by comparing the estimates of the mutation rates of both groups. If H2a is true, the mutation rate, μ, should be lower for FTF groups than CMC groups.

We can measure diversity (Δ) as the distance between the solutions, or points, in the solution space. Genetic algorithm research has concerned itself extensively with the concept of Hamming distance [7]. Mitchell [11] defines Hamming distance as the number of locations or genes at which the corresponding values or bits differ. Other such distance measures are possible, however Hamming distance represents the simplest and most widely used distance measure for complex search spaces in genetic algorithm literature. We will compute the average Hamming distance for each group by computing the Hamming distance between every pair of solutions in the group and summing up all of the distances. This sum is then divided by the number of solution pairs in the group to create an average diversity level for each group. We call this measure Δ. Diversity is examined as it also provides insight into variation among the different solutions proposed by the groups.

H2b: FTF groups are likely to have a lower diversity than CMC groups.

As Δ becomes small, χ should also become small for FTF groups. As there is less diversity within the group, the crossover rate, or the rate at which "parts" of proposals or ideas are exchanged, will become small, as most of the proposals are already identical. As CMC groups experience less diversity, there will be a higher rate of exchange of proposals, or at least components of proposals. Correspondingly, Δ and χ should both be larger for CMC groups.

H2c: If there is little or no diversity within the groups, FTF groups are likely to experience a lower rate of exchange of ideas or proposals than CMC groups.

We will test this hypothesis by comparing the crossover rates, χ, between FTF and CMC groups. Given that diversity is lower for FTF groups as proposed in H2b, χ should be lower for FTF groups than for CMC groups.

B. Experimental Data

To test and validate our model, we use data provided by Barkhi [1]. Barkhi's experiments examined the effects of various factors on the outcome of GDSS problem-solving tasks. They considered a mixed-motive task by which group members had to coordinate the final solution in face of conflicting pay-off information. They constructed a group where each member represented a different department within a simulated manufacturing environment, the departments being labeled as production, purchasing and marketing. Some of the groups in the study were comprised of the three members labeled as above and the others were comprised of four members (the previous three plus a designated "leader" who had override power on all decisions made within the group). The group was assigned a combinatorial problem with a calculated payoff for each member.

The experimental groups were provided the following problem to solve. Each group member represented a different functional manager in a simulated manufacturing firm. Each group member was provided cost and revenue data for a set of twenty customer orders. As each group member represented a different "department" with varying cost data, there were conflicts concerning resource allocation built into the experimental problem. Due to different capacity constraints among departments, not all of the orders could be filled. The group members also had to decide how much effort (given cost and revenue data for varying effort levels) to expend in filling a particular customer order. Therefore, the group members had to solve a type of multi-objective knapsack problem.

Three research variables were studied. These were group composition (leader vs. no leader), proximity (face-to-face vs. geographically distributed) and member incentive structure (local vs. global). Barkhi tested two different incentive schemes. One scheme, local incentive, rewarded each manager based on how well the manager controlled actual costs compared to projected costs. The other scheme, global incentive, rewarded each manager based on an equal percentage of organizational bonus, corresponding to organizational profit.

C. Model Details

This section presents the problem-specific details of the research model. The encoding of the strings, population sizing, fitness function and GA operator implementation will be described.

Each string in the population will represent a set of orders to be filled as proposed by a manager or the leader in a specific generation. Each string is composed of twenty binary digits, each representing the inclusion (or exclusion) of a customer order by a one (or zero).

A population consists of a number of solutions. The number varies from episode to episode, but there is no "natural" demarcation of these episodes for groups using GDSS. Hence, GDSS supported groups can be modeled as having a dynamic population size. We propose four different schemes for modeling this: Leader-Influenced, Peer-Influenced, Fixed-Two and Fixed-Four.

The Leader-Influenced population scheme is implemented for groups having a leader and is modeled as follows. The population size expands (or contracts) depending upon the frequency of leader interaction with the three departmental managers, with more leader interaction corresponding to smaller population sizes. Each generation will be delineated by the proposal of a solution by the leader. For example, the marketing manager proposes a solution, which is countered by the production manager. The leader makes a proposal after the production manager. Since the leader has suggested a solution, this suggestion marks the entirety of the population and its size is three. If the purchasing manager then suggests a solution, followed by another solution by the leader, that population size is two. It is worthwhile to note that this strategy relies not on the timing of the solutions but on the interaction of the different group member's solutions (ideas). Theoretically, it might be preferable to close each generation after input has been made by all three functional managers. However, due to group interaction dynamics, it is possible that one or more managers might engage in freeloading behaviors (especially in global incentive groups as mentioned in a previous section) and artificially influence the creation and sizing of generations.

The Peer-Influenced population scheme is similar to the Leader-Influenced population scheme except that the Peer-Influenced scheme treats each group member (each function manager and leader, if present) as having ideas of "similar enough to equal" weight. Whenever a solution from a different manager is presented, the current generation is closed. For example, the marketing manager proposes a solution, then another solution immediately afterwards. The production manager then proposes a solution. This marks the end of the generation and the population size is three. For groups with a leader, the leader is simply regarded as another functional manager. However, several issues are raised. The Leader-Influenced scheme was proposed due to the belief that leaders exert a different type of influence over the decision-making process than do peers. Therefore, some account of this variance in influence should be taken in forming the generations. Also, the type of leadership style exhibited by each leader needs to be examined. A cursory examination of the data indicates that some leaders interact with their groups far more than other leaders. This finding indicates that some leaders are adopting a "hands-on" management style as opposed to a "hands-off" management style. This issue most likely will not be adequately addressed nor resolved in this research project so we will defer it to the area of future research.

Two more schemes are used experimentally, for comparison purposes. The Fixed-Two scheme places every two consecutive proposals into a population and then the proper operations are carried out. The Fixed-Four scheme places every four consecutive proposals together into a generation. This scheme acts to increase the diversity levels from the Fixed-Two scheme.

The fitness functions can be viewed as an implementation of the social welfare functions described in economic agency theory literature. Luce and Raiffa define social welfare function as "..a rule which associates to each profile of preference orderings (i.e., to each element of $\Re^{(n)}$) a preference ordering for society iteself." ([10], pp. 332). More specifically, the fitness function used is to be an additive utilitarian social welfare function as described in Moore [12]. The fitness function used by the GA is a linear combination of the various individuals' utility functions and the corporate utility function, captured in the leader's utility function. The fitness function is therefore expressed as a combination of the group member's utility function and the leadership function, where the weights on each function are varied. This weighted combination of utility functions represents the social welfare function described earlier. These weights are experimentally varied from 0.0 to 1.0, where 0.0 represents exchanging ideas or solutions with local incentive and no corporate influence and 1.0 represents global incentive and the strongest corporate influence. Restated, a corporate weight of 0.0 corresponds to groups operating under local incentive and a corporate weight of 1.0 corresponds to groups operating under global incentive. The spectrum of weights from 0.0 to 1.0 represents the strengthening of corporate influence on the groups' incentive structure.

Selection is implemented as "roulette-wheel" selection, due to its simplicity. Crossover is implemented using single-point crossover, again due to the simplicity of the scheme. An illustration of single-point crossover on two ideas, A and B is illustrated in Figure 1 below. The vertical line | denotes the crossover site. The portions of the strings to the left are exchanged. Finally, uniform mutation is used, with very low mutation rate settings (0.001,0.01) as is common in the GA literature. Figure 2 illustrates the use of uniform mutation where the underlined bit is the one that is mutated.

Before crossover
Idea A: 10010011|1010010010
Idea B: 01001000|1001010010
After crossover
Idea A' : 10010011|1001010010
Idea B' : 01001000|1010010010

Figure 1: Single-point crossover implementation

Before mutation
Idea A' :100100111001010010
After mutation
Idea A": 101100111001010010

Figure 2: Uniform mutation implementation

D. Methodology

We used the Markov model proposed by Nix and Vose [13] as the basis for our determination of the likelihood function for the probabilities of each group's actual decision path through the search space. The maximum likelihood estimates (MLEs) for these paths were calculated over all possible values of mutation (0, 0.5) and crossover (0, 1.0) within 3-digit precision. Therefore, we estimated the actual mutation and crossover rates for each group, assuming each group acted like a simple GA.

III. Results

The main hypothesis, that groups using GDSS behave like GAs, was tested by comparing the maximum likelihood ratios of the path probabilities of the estimated parameters (μ and χ) to the probability of these paths under a random search (μ =0.5 and χ=0) using the Wilcoxon matched-pairs signed-ranks test. For each of the parameter hypotheses, we used a one-tailed t-test assuming unequal variances on the sample means at $\alpha = 0.05$ to test differences in the populations for each parameter.

The table below (Table 1) shows that there is strong support for our main hypothesis. The test statistic, w_α, is calculated via the procedure outlined in Conover [2]. We do not reject our hypothesis if the critical value, T, is greater than $w_{1-\alpha}$ [2]. The critical values for all experimental conditions are much greater than the test statistics indicating a very low probability of error.

Our first parameter hypothesis, H2a, stated that FTF groups are likely to have less radical or extreme proposals than CMC groups. In order to test this hypothesis, we compared the mutation rate, μ, between FTF and CMC groups. For this hypothesis to not be rejected, FTF groups must have a smaller μ than CMC groups. We examined several variations. For all experimental conditions, this hypothesis was rejected at $\alpha = 0.05$. The overall mean for FTF groups is lower (0.023) than the overall mean for CMC groups (0.025) but not significantly so.

Table 1: Test results for main hypothesis

Incentive Weighting Scheme		
		0.0 - 1.0
Population Sizing Scheme	**A**	T= 990 Do Not Reject $w_{.95}$=635.9
	B	T = 1081 Do Not Reject $w_{.95}$=691.1
	C	T = 1128 Do Not Reject $w_{.95}$=719.4
	D	T= 990 Do Not Reject $w_{.95}$=635.9

The next hypothesis posed compares the level of diversity (difference in solutions as measured by the Hamming distance) between FTF and CMC groups. Hypothesis H2b stated FTF groups are likely to have a lower diversity value (as measured by Δ) than CMC groups. This hypothesis was rejected at $\alpha = 0.05$. The difference in the means for the two groups are significant, however the CMC groups have the lower level of diversity than the FTF groups.

Table 2: Results of statistical tests on communication channel hypotheses

Hypothesis	**FTF Mean**	**CMC Mean**	**Results**
H2a	0.0233	0.0252	0.3540 Reject H2a
H2b	1.2767	0.8596	0.0418 Reject H2b
H2c	0.0197	0.0694	0.0258 Do Not Reject H2c

Our final hypothesis comparing FTF groups with CMC groups relates the crossover rate, χ, to the diversity rate. Hypothesis H2c stated if Δ is close to zero, FTF groups are likely to behave like a GA having a lower χ than CMC groups. The mean Hamming distance for both FTF and CMC groups is small, on average about one bit difference per pair of solutions. The one-tailed t-test results for the crossover rates from all the experimental conditions are presented in Table 2. We do not reject hypothesis H2c at $\alpha = 0.05$.

IV. Conclusions

As discussed in the previous section, we conclude that groups using GDSS do act like a simple genetic algorithm using selection, crossover and mutation. There exists strong and compelling evidence that the search undertaken by the GDSS groups did not behave like a random search, but sampled solutions according to fitness and exchanged and altered parts of solutions at the rates estimated from the data. In this study, we tested two parameters, χ and μ, and their effects on communications channel. Linking the GA parameters to the communications channel variable was unsuccessful. H2a had the correct relationship, but it was not significant. More data might illuminate this. For H2b, where our results were contrary to our hypothesis, there might be interaction effects at work. These effects were not considered for this particular study.

It is readily apparent that more data sets are required to carry out robust and accurate testing of this particular model, since most of our results came out as we anticipated but were not statistically significant. There are several issues to consider regarding the genetic algorithm used to model groups using GDSS. We believe that our selection operator is not robust enough to adequately describe the actual process of sampling ideas based on fitness function information. Rank or tournament selection operators might model real-world decision-making more accurately. Perhaps even more obviously, the single-point crossover operator employed does not represent the actual mating process of proposed solutions during the brainstorming and negotiation phases of this particular task. We believe multi-point or uniform crossover might more accurately depict the idea-exchange process.

As previously discussed, we varied both the population sizing and incentive weighting schemes. The best overall population-sizing scheme seemed to be the Peer-Influenced scheme. It appeared to perform better than the Leader-Influenced scheme as the Leader-Influenced scheme did not account for wide variations in leadership style. It was made apparent by visual inspection of the data that some leaders were quite active in proposing solutions where others were more passive, that is only proposing a solution towards the end of the session. This affected the population sizes and the number of generations created. We believe a hybrid scheme where leadership style is somehow incorporated by weighting the leader’ s proposed solutions depending on activity level for determining generations would interesting to study. The worst scheme was the Fixed-Two scheme. Considering this scheme was designed to be somewhat arbitrary and would have little diversity in sampling solutions for the next generation, the scheme’ s poor performance makes sense. The Fixed-Four scheme performed better, as the population sizes were larger, thus improving outcomes due to a larger number of solutions to sample from.

The incentive weighting scheme tends to favor higher weights corresponding to increasing corporate influence. The implications of this finding are unclear at this point. Due to the different treatment conditions and possible interaction effects, there is no obvious interpretation of these results.

Finally, we can address the utility in viewing groups using GDSSs as GAs. There exists a large body of heuristic knowledge in the GS literature that could be used to construct group sizes and characteristics as related to mutation and crossover (although it should be emphasized that much more work is needed to understand the nature of this relationship). From GA theory, we can estimate bounds on the time required to evolve good ideas given various combinations of task type, incentive structure, communication channel, etc.

V. Future Research

This project provides many opportunities for future research. We would like to improve the GDSS model by incorporating the more expressive genetic algorithm operators described in the previous section. We would also like to include features of other models, particularly the economic model for GDSS as proposed by Gavish and Kalvenes [6]. Obviously, we require more data from actual GDSS experiments to further validate our model, especially from experiments which further examine the variables studied in Barkhi [2].

There are also future research implication of particular interest to GDSS researchers. Not only are there other GDSS configurations of interest, but also meetings that take place different-time different-place, meetings that occur over an extended period of time and meetings that take place within virtual organizations [9] to name a few limited scenarios. Other task types and GDSS variables can be considered. We also feel that this model eventually can be applied to non-GDSS supported groups.

References:

[1] Barkhi, R. (1995) "An Empirical Study of the Impact of Proximity, Leader and Incentives on Negotiation Process and Outcomes in a Group Decision Support Setting," Doctoral Dissertation, The Ohio State University.

[2] Conover, W. J., (1971) *Practical Nonparametric Statistics*, John Wiley & Sons Inc., NewYork.

[3] DeSanctis, G. and R. B. Gallupe (1987) "A Foundation for the Study of Group Decision Support Systems," *Management Science*, Vol. 33 no. 5, pp. 589-609.

[4] Gavish, B., Gerdes, Jr., J., and Sridhar, S. (1994) "CM^3, Looking into the Third and Fourth Dimensions of GDSS," *Information and Collaboration Models of Integration*, ed. S. Y. Nof, Kluwer Academic Publishers, The Netherlands, pp. 269-299.

[5] Gavish, B., Gerdes, Jr., J., and Sridhar, S. (1995) "CM^3 – A Distributed Group Decision Support System," *IIE Transactions*, Vol. 27, pp. 722-733.

[6] Gavish, B. and J. Kalvenes (1998) "An Economic Model of Group Decision Support Systems," *Information Systems Research*, Forthcoming.

[7] Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, Mass.

[8] Hirokawa, R. and D. Johnson (1989) "Toward a General Theory of Group Decision Making Development of an Integrated Model," *Small Group Behavior*, Vol. 20, No. 4, pp. 500-523.

[9] Jarvenpaa, S. L. and B. Ives (1994) "The Global Network Organization of the Future: Information Management Opportunities and Challenges," *Journal of Management Information Systems*, Vol. 10, no. 4, pp. 25-57.

[10] Luce, R. D. and H. Raiffa (1957) *Games and Decisions Introduction and Critical Survey*, Dover Publications, Inc., New York.

[11] Mitchell, M. (1996) *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, MA.

[12] Moore, J.C., Rao, H.R., Whinston, A., Nam, K. and T.S. Raghu (1997) "Information Acquisition Policies for Resource Allocation Among Multiple Agents," *Information Systems Research*, Vol. 8, no. 2, pp.151-170.

[13] Nix, M.A., and M.D. Vose (1992), "Modeling Genetic Algorithms with Markov Chains," *Annals of Mathematics and Artificial Intelligence*, 5, pp. 79-88.

[14] Nunamaker, J.F., Dennis, A. R., Valacich, J. S., Vogel, D. R., and J. F. George (1991) "Electronic Meeting Systems to Support Group Work," *Communications of the ACM*, Vol. 34 no. 7, pp.40-61.

[15] Sia, C.-L., Tan, B. C. Y., and K.-K. Wei (1996) "Will Distributed GSS Groups Make More Extreme Decisions? An Empirical Study," *Proceedings of the Seventeenth International Conference on Information Systems*, Eds. J. I. DeGross, S. Jarvenpaa and A. Srinivasan, Cleveland, OH, pp. 326-338.

[16] Valacich, J. S. and A. R. Dennis (1994) "A Mathematical Model of Performance of Computer-Mediated Groups during Idea Generation," *Journal of Management Information Systems*, Vol. 11, no. 1, pp. 59-72.

[17] Vose, M.D. (1998), *The Simple Genetic Algorithm: Foundations and Theory*, The MIT Press, Cambridge, MA.