CERIAS Tech Report 2001-123

Evolution in Groups: A Genetic Algorithm Approach to Group Decision Support Systems

by J Rees, G Koehler
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Information Technology and Management 3, 213–227, 2002

Evolution in Groups: A Genetic Algorithm Approach to Group Decision Support Systems *

JACKIE REES ** jrees@mgmt.purdue.edu
Krannert Graduate School of Management, Purdue University, West Lafayette, IN 47907-1310, USA

GARY KOEHLER koehler@ufl.edu
Decision and Information Sciences, Warrington College of Business, University of Florida, Gainesville, FL 32611, USA

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. For these types of GDSS tasks, we propose to model the information exchange and convergence toward a solution 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 current population 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.

Keywords: group decision support systems, genetic algorithms, computational modeling

1. Introduction

Groups of individuals meeting to solve particular problems can be viewed as searching for a solution within some sort of solution space [6,16]. This search space is most likely highly complex, otherwise the collective expertise of a group would most likely not be required. 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 groups faced with these tasks, we propose modeling the decision making process as a simple genetic algorithm.

Researchers have 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 or negotiating session are subject to adaptation is not a new idea but has not been formally incorporated into analytical models for GDSS.

* An earlier version of this paper was presented at the *32nd Hawaii International Conference on System Sciences*, 5–8 January 1999.
** Corresponding author.

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.

2. 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.

2.1. Analytical models for GDSS

Group decision support systems are designed to support group decision-making through specialized software, hardware and decision support tools. GDSS have been defined 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 [3]. 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.

One of the earliest computational models for group decision making under GDSS was a simple mathematical model of electronic brainstorming [17]. This model presents GDSS brainstorming as the ideas generated by a group of individuals, each working alone, accounting for process losses and process gains [14]. In other words, "... group performance is a function of individual performance minus process losses plus process gains", [17, p. 64]. The model was one of the earliest models to provide analytical insight into a particular GDSS process. Specifically, the model was a regression model that expressed the number of unique ideas generated by the group during a brainstorming session as a function of the number of group members. 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 [4,5] where brainstorming and other GDSS activities were closely examined. One of the important features of this economic 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 this 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.

2.2. 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 points, rather than the traditional single point, as is the case in linear programming or gradient descent. Such points are often referred to as strings or chromosomes [7]. These points are candidate solutions to the problem, encoded as sequences of digits (bits) [11]. These points, referred to as strings, 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 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. Details on GAs can be found 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. Mutation is often referred to as an "exploration" operator in this sense. The function of the mutation operator in a mathematical sense is to direct the search away from local optima.

3. 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 behind the genetic algorithm to better describe the information exchange, negotiation and convergence to a solution 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 this idea proposal or generation process as 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. At each point in time, the current proposed solutions are represented by a population of strings. Selection, crossover and mutation operate on these strings as described above. As the generations evolve, the genetic algorithm tends to find better and better solutions.

We have chosen a simple genetic algorithm as the basis of this model for several reasons. First, like the group process we hope to mimic, GAs are adaptive, with the population of proposed solutions changing over time, in response to the environment, the fitness function and other constraints. This captures the adaptation [8] and the trigger phenomenon [6] discussed earlier in the paper.

The second reason for using a GA is that formal, mathematical theory has been developed to describe the expected behavior of the simple genetic algorithm [18]. Provided the GA adequately mimics a given GDSS process, a large number of properties can be computed for the GDSS by computing these for the GA. The exact expected behavior of a GA is represented by a Markov chain that is a function of the various GA parameters [13,18]. With this chain many properties can be computed, such as the expected first passage times from a set of states to a target set of states. In our setting, the states of the Markov chain are populations of strings representing solutions to a problem; the interpretation of the first-passage times, first-passage probabilities, and other computable items gives us many useful values.

For example, the expected time to see an optimal solution is the expected time of the first passage to any population containing a string that maximizes the GA fitness function (i.e., to a solution to the underlying GDSS problem). The expected time to a desired level of consensus is the expected first-passage time to any population consisting of an appropriate proportion of identical strings. The probability of a trigger phenomenon can be computed using transition probabilities to populations having little in common with the current population. Similarly, the expected number of new "ideas" can be computed.

Although *a priori* we do not link specific variables and environmental pressures thought to influence the GDSS process, to our GA model, one may be able to relate such factors *a posteriori* perhaps through regression. Then, knowing how GA parameters influence various performance measures from formal GA theory, one could optimize these by altering the GDSS design and usage. Even if one could not directly optimize these performance measures, there exists a large body of heuristic knowledge available in the GA community for the GA to improve its search capabilities, which could be used indirectly to set GDSS design and usage through these GDSS-GA parameter relations.

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. The GA could be used in conjunction with a GDSS process to supply ideas. It may be used in place of an actual group.

4. Research questions

The underlying assertion for this work is that groups using GDSS act like simple genetic algorithms using selection, crossover and mutation. Two types of GDSS use are considered for comparison, Face-To-Face (FTF) groups and geographically-distributed, or Computer-Mediated-Communication (CMC) groups. Note in the following that χ stands for "crossover rate" and μ stands for "mutation rate". Our main assertion is:

Assertion. 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 (see [7] for details). Furthermore, we can examine this assertion by formulating the hypothesis H1 below. Given an observed sequence of populations produced by a group, we can estimate GA mutation and crossover parameters (μ and χ) which give a high probability that the equivalent Markov chain would produce the same sequence of populations. We call the probability of seeing a particular sequence of populations a path probability. A GA produces a sequence of solutions by applying various operators on the current set of proposed solutions. Different sequences have different probabilities of being realized (which an equivalent Markov chain captures). Presumably, a group has internal factors that shape their sequence of proposed solutions. They may be following an opportunistic strategy (like a greedy algorithm) or something else. In the absence of specific knowledge of the group's process, we assume an uninformed prior on the mechanism and consider it essentially random. Future work could consider testing other plausible group mechanisms.

H1: The maximum likelihood ratios of the path probabilities of the estimated parameters (μ and χ) will differ from the probability of these paths under a random search, using the data generated from the described GDSS sessions.

This hypothesis will be tested using the methodology described in section 5.

One of the issues facing group work is the effect of social and political forces that possibly affect the quality of the group's decision-making. We hypothesize that 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. 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, p. 328, 17]. In other words, CMC groups can be considered more explorative of the solution space than FTF groups. The potential impact of this hypothesis is if we better understand the degree of search space exploration by particular types of groups, we might be able to further manipulate the search in hopes of finding more favorable solutions.

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 will propose less explorative solutions than CMC groups as measured by the maximum likelihood estimate of each group's mutation rate.

We will test this hypothesis by comparing the estimates of the mutation rates of both groups, as discussed in section 1. Part C, the mutation operator acts as an "exploration" operator. If H2a is true, the mutation rate, μ, should be lower for FTF groups than CMC groups.

Solution diversity within the population is examined as it provides insight into variation among the different solutions proposed by members of the groups. As stated above, FTF groups are likely to generate and explore similar ideas than CMC groups, likely resulting in lower diversity for FTF groups.

We can measure diversity (Δ), of a population, as the average distance between the points of the population. Genetic algorithm research has concerned itself extensively with the concept of Hamming distance [7]. Hamming distance is defined [11] 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.

H2b: FTF groups will have a lower diversity than CMC groups as measured by each group's average Hamming distance.

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 will experience a lower rate of exchange of ideas or proposals than CMC groups as measured by the maximum likelihood estimate of each group's crossover rate.

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.

5. Experimental data

To test and validate our model, we use data provided by actual GDSS experiments [1]. These experiments examined the effects of various factors on the outcome of GDSS problem-solving tasks. The factors included communication channel, leadership presence and incentive structure. Outcome was measured in terms of both group-member satisfaction with the process and overall profitability of the group decision, explained below. The investigators considered a mixed-motive task by which group members had to coordinate the final solution in face of conflicting pay-off information. Groups were constructed where each member represented a different department within a simulated manufacturing environment, the departments being labeled as production, purchasing and marketing. Data was collected from a total of forty-eight groups. 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. Each of the twenty customer orders was a combination of four different products, with varying quantities of each product specified. 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). The study's investigators 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.

6. 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 from the final order set by a one (or zero).

A population consists of a number of solutions. Typical GA applications call for a fixed number of strings in each population. However, there seems to be no logical way of "fixing" the number of solutions proposed by members of the group at any particular time interval. The number of solutions generated varies over the course of the GDSS session. 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 (scheme A) 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 section 5) and artificially influence the creation and sizing of generations.

The Peer-Influenced population scheme (scheme B) 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 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" leadership style as opposed to a "hands-off" leadership 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 population schemes are used experimentally, for comparison purposes. The Fixed-Two population scheme (scheme C) places every two consecutive proposals into a population and then the proper operations are carried out. The Fixed-Four population scheme (scheme D) places every four consecutive proposals together into a generation. This population scheme acts to increase the diversity levels from the Fixed-Two scheme. Figure 1 displays the population placement of a typical proposed solution sequence according to each of the previously discussed schemes. The source of each proposed solution is indicated by "L" for leader, "M" for marketing, "P" for production or "R" for procurement.

The fitness functions can be viewed as an implementation of the social welfare functions described in economic agency theory literature. A social welfare function is defined as "... a rule which associates to each profile of preference orderings ... a preference ordering for society itself" [10, p. 332]. More specifically, the fitness function used is to be an additive utilitarian social welfare function [12].

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 2. The crossover site is indicated by ^. The portions of the strings to the left are exchanged. Finally, uniform mutation is used, with very low mutation rate settings (from 0.001 to 0.01) as is common in the GA literature. Figure 3 illustrates the use of uniform mutation where the underlined bit is the one that is mutated.

7. Methodology

The Markov model [13] was used as the basis for the determination of the likelihood function for the probabilities of each group's actual sequence of proposed solutions 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 from the GDSS experiments [1], assuming each group acted like a simple GA.

Proposed solution sequence	
M:	00100011010001000011
P:	10100010010010001011
M:	00010011010000000011
L:	00100011010000000011
L:	00100011010001000011
P:	00100111010010000000
P:	00100011010001000011

Leader-Influenced (scheme A)	*Peer-Influenced (scheme B)*
Population 0:	Population 0:
00100011010001000011	00100011010001000011
10100010010010001011	10100010010010001011
00010011010000000011	
00100011010000000011	
Population 1:	Population 1:
00100011010001000011	00010011010000000011
	00100011010000000011
Population 2:	Population 2:
00100111010010000000	00100011010001000011
00100011010001000011	00100111010010000000
	Population 3:
	00100011010001000011

Fixed-Two (scheme C)	*Fixed-Four (scheme D)*
Population 0:	Population 0:
00100011010001000011	00100011010001000011
10100010010010001011	10100010010010001011
	00010011010000000011
	00100011010000000011
Population 1:	Population 1:
00010011010000000011	00100011010001000011
00100011010000000011	00100111010010000000
	00100011010001000011
Population 2:	
00100011010001000011	
00100111010010000000	
Population 3:	
00100011010001000011	

Figure 1. Population sizing schemes.

Before crossover
Idea A: 10010011^1010010010
Idea B: 01001000^1001010010
After crossover
Idea A′: 10010011^**1001010010**
Idea B′: 01001000^**1010010010**

Figure 2. Single-point crossover implementation where ^ denotes the crossover site.

Before mutation
Idea A′: 10<u>0</u>100111001010010
After mutation
Idea A″: 10**<u>1</u>**100111001010010

Figure 3. Uniform mutation implementation where _ denotes the mutated bit.

Table 1
Test results for main hypothesis.

Population sizing scheme	Test results
Leader-Influenced	$T = 990$, do not reject, $w_{0.95} = 635.9$
Peer-Influenced	$T = 1081$, do not reject, $w_{0.95} = 691.1$
Fixed-Two	$T = 1128$, do not reject, $w_{0.95} = 719.4$
Fixed-Four	$T = 990$, do not reject, $w_{0.95} = 635.9$

8. Results

The main assertion, that groups using GDSS behave like GAs, was examined by testing hypothesis H1. H1 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 ($\mu = 0.5$ and $\chi = 0$) 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 1 shows that there is strong support for the main hypothesis. The test statistic, w_α, is calculated via the procedure outlined in [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. Therefore, we conclude that there is strong support for considering groups using GDSS as behaving more like a genetic algorithm with the estimated parameter settings than as a random process.

Our first parameter hypothesis, H2a, stated that FTF groups 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,

Table 2
Results of statistical tests on communication channel hypotheses.

Hypothesis	FTF mean	CMC mean	Results
H2a	0.0233	0.0252	0.3540, not supported
H2b	1.2767	0.8596	0.0418, not supported
H2c	0.0197	0.0694	0.0258, supported

FTF groups must have a smaller μ than CMC groups. 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. There could be several reasons for this lack of significance. First, more data is needed to robustly test this hypothesis over different experimental situations. Second, other factors, such as the interaction between other variables could be clouding the results. Finally, the specific GA model (roulette-wheel selection, single-point crossover and uniform mutation) used in this study might not be sensitive enough to capture the exploration processes involved in GDSS use.

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 have a lower diversity value (as measured by Δ) than CMC groups. This hypothesis was rejected at $\alpha = 0.05$. The differences in the means for the two groups are significant, however the CMC groups have the lower level of diversity than the FTF groups. The results of this hypothesis seem contrary to the findings of the previous hypothesis. This finding indicates that we are far from complete in our understanding of the GDSS group search process.

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 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 are presented in table 2. We do not reject hypothesis H2c at $\alpha = 0.05$. The outcome of this test indicates that there is less exchange of new or different information between group members during GDSS use in FTF groups than in CMC groups. This result not only affirms previous research but also quantifies the difference in information exchange.

9. Conclusions

As discussed in section 8, we conclude that groups using GDSS appear to behave more like a simple genetic algorithm using selection, crossover and mutation than as a random process. 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. This finding indicates that a new approach to computational study of GDSS is now possible. By estimating the search parameters for particular types of groups, based on previous data, researchers and practitioners can adjust environmental settings, such as communication channel or reward in order to improve group performance. Additionally, if the estimated crossover and mutation rates were less than optimal for a specific task, the group dynamics could be manipulated to better manage the group decision-making process.

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 ambiguous at best. H2a had the correct relationship, but it was not significant. More data might better illuminate this relationship. For H2b, where our results were contrary to our hypothesis, more work needs to be done in understanding the relationships between solution diversity, exploration and exploitation. H2c was found to be significant, indicating the potential to quantify the rate of information exchange between group members. This information can be used to make adjustments in FTF groups, for example the design of improved incentive or reward schemes to allocate better rewards for higher rates of information exchange.

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 not all were 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 the population-sizing scheme. The best overall population-sizing scheme seemed to be the Peer-Influenced scheme. It appeared to perform better in terms of parameter variance 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 research into a hybrid scheme where leadership style is incorporated by weighting the leader's proposed solutions according to activity level for determining generations could be fruitful. 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.

Finally, we can address the utility in viewing groups using GDSS 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.

10. 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 section 9. As a reviewer pointed-out, other optimization methods or other meta-heuristic search models might be as appropriate as GAs in modeling the behavior of group decision processes. It would be worthwhile to examine the applicability of other methods such as tabu search, simulated annealing, math programming models, and various hybrids, to the problem of group decision-making, to see if there is any improvement over the current GA-based approach. Our focus on GAs stems largely from its adaptive nature, which has a clear, strong parallel with actual group activities – a point previously elaborated by others [8]. However, one could imagine other driving forces of group decision processes, such as exploration of similar (close) ideas and recall and avoidance of ideas already explored, which might be better captured by tabu search or another meta-heuristic.

In future work we would also like to include features of other models, particularly the economic model for GDSS [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 [1].

There are also future research implications 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] R. Barkhi, 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 (1995).

[2] W.J. Conover, *Practical Nonparametric Statistics* (Wiley, New York, 1971).

[3] G. DeSanctis and R.B. Gallupe, A foundation for the study of group decision support systems, Management Science 33 (1987) 589–609.

[4] B. Gavish, J. Gerdes, Jr. and S. Sridhar, CM^3, looking into the third and fourth dimensions of GDSS, in: *Information and Collaboration Models of Integration*, ed. S.Y. Nof (Kluwer Academic Publishers, The Netherlands, 1994) pp. 269–299.

[5] B. Gavish, J. Gerdes, Jr. and S. Sridhar, CM^3 – A distributed group decision support system, IIE Transactions 27 (1995) 722–733.

[6] B. Gavish and J. Kalvenes, Economic issues in group decision support systems, in: *Proceedings of the 1st INFORMS Conference on Information Systems and Technology*, eds. H. Pirkul and M. Shaw (INFORMS, Washington, DC, 1996) pp. 18–27.
[7] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison Wesley, Reading, MA, 1989).
[8] R. Hirokawa and D. Johnson, Toward a general theory of group decision making development of an integrated model, Small Group Behavior 20 (1989) 500–523.
[9] S.L. Jarvenpaa and B. Ives, The Global Network Organization of the future: Information management opportunities and challenges, Journal of Management Information Systems 10 (1994) 25–57.
[10] R.D. Luce and H. Raiffa, *Games and Decisions Introduction and Critical Survey* (Dover, New York, 1957).
[11] M. Mitchell, *An Introduction to Genetic Algorithms* (The MIT Press, Cambridge, MA, 1996).
[12] J.C. Moore, H.R. Rao, A. Whinston, K. Nam and T.S. Raghu, Information acquisition policies for resource allocation among multiple agents, Information Systems Research 8 (1997) 151–170.
[13] M.A. Nix and M.D. Vose, Modeling genetic algorithms with markov chains, Annals of Mathematics and Artificial Intelligence 5 (1992) 79–88.
[14] J.F. Nunamaker, A.R. Dennis, J.S. Valacich, D.R. Vogel and J.F. George, Electronic meeting systems to support group work, Communications of the ACM 34 (1991) 40–61.
[15] C.-L. Sia, B.C.Y. Tan and K.-K. Wei, Will distributed GSS groups make more extreme decisions? An empirical study, in: *Proceedings of the 17th International Conference on Information Systems*, eds. J.I. DeGross, S. Jarvenpaa and A. Srinivasan (Cleveland, OH, 1996) pp. 326–338.
[16] H.A. Simon, *The New Science of Management Decision* (Prentice-Hall, Englewood Cliffs, NJ, 1977).
[17] J.S. Valacich and A.R. Dennis, A mathematical model of performance of computer-mediated groups during idea generation, Journal of Management Information Systems 11 (1994) 59–72.
[18] M.D. Vose, *The Simple Genetic Algorithm: Foundations and Theory* (The MIT Press, Cambridge, MA, 1998).