ROBOT SOCRATES: CONTRADICTION IDENTIFICATION WITH MINIMUM QUESTIONING

by

Richard Marmorstein

2014

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

I develop a concept of computer-assisted Socratic dialogue, aimed to enable productive and efficient argumentation. The goal is a framework for argumentation that is as rigorous as a formal debate, yet as convenient as an online quiz. A human specifies questions and answer choices, and designates certain sets of answer choices as contradictory. The computer's task is to end the dialogue asking as few questions as possible, by eliciting a set of answers from a questionee that either includes a contradiction, or eliminates the possibility of any contradiction. I formalize this minimum questioning problem, and prove it is NP-hard. I then analyze the problem in terms of a trade-off between asking questions that are most likely to achieve immediate progress towards termination, and asking questions which are most likely to yield knowledge which will lead towards greater progress towards termination in the future. I develop a greedy algorithm that maximizes a multi-objective utility function embodying this trade-off, and evaluate its performance on a set of randomly generated dialogues and questionees. Results suggest that favoring future progress over immediate progress is a better strategy only in contrived cases. The algorithm is able to adapt well between between contradictory and non-contradictory cases, requiring on average only a fifth as many questions as random question selection in the contradictory case, and less than half as many questions in the non-contradictory case.

Chapter 1

INTRODUCTION

Argumentation has played a key role in human history ever since the very first debate: the Serpent vs. Eve. It remains important today, both as an essential activity for such institutions as courts and legislatures, but also as a form of recreation. Recreational argumentation, which may take place in person or in writing, does more than entertain its participants–it helps spread ideas and shape public opinion.

I am interested in the productivity of argumentation. In a productive argument, there is learning. At least one participant comes to a greater understanding of an opposing view, or becomes more critical of his own. This does not always happen in practice. In the worst case, an argument devolves into a mere exchange of sarcasm and insults, but even an argument that remains civil can drag on, with participants repeating themselves, misunderstanding positions, and failing to address the point of disagreement.

Some ways of arguing encourage productivity. The ancient Greek philosopher Socrates was well known for his style of dialogue. Today, the phrase "Socratic dialogue" can refer to a number of practices. I investigate the pattern of argumentation discussed by Caminada, who also calls it "hang yourself argumentation" [**?**]. In this type of Socratic dialogue, one participant, the questioner, argues via asking the other participant a sequence of questions. His goal is to lead the questionee into giving contradictory answers, in which case she must either retract her entire stance, or enough answers to eliminate the contradiction. Possibly, the questionee will evade contradiction by delivering an unanticipated answer, leading to another productive outcome. This pattern of argumentation directs debate to points of disagreement, and encourages productive debate.

But there are disadvantages to this form of argumentation. It is difficult to produce questions that will lead an opponent into contradiction, especially ad hoc. Even Socrates himself may not have been talented enough: scholars believe that accounts of his dialogue may be largely pieces of fiction composed by his student, Plato, who did not face time constraints in selecting questions [**?**]. Written-out dialogues also have shortcomings—questions must be selected for the generic reader, and cannot be tailored for each questionee's particular beliefs. The reader of such a dialogue must spend time considering questions intended to debunk beliefs that she does not hold. This may be edifying, but it is not productive in the sense of bringing the reader to a critical re-examination of her own views.

In this thesis, I introduce a concept of computer-assisted Socratic dialogue, which occupies a middle ground between pre-written and ad hoc dialogue. In my model, a human questioner first composes a series of questions, as in a pre-written dialogue. The questioner also composes answer choices to describe beliefs he deems potential questionees are likely to hold regarding those questions. Last, he identifies groups of answers as *contradictions*—beliefs that he judges to be incompatible with each other. A computer then presents the dialogue to questionees. The computer chooses an initial question to ask, and the questionee indicates an answer. The computer proceeds, choosing which question to ask next based on which answers the questionee has already selected. The process ends when the questionee has delivered a contradictory set of answers or when no contradiction can possibly be reached. As the computer goes through this process with more and more questionees, it learns which questions to ask first in order to bring the dialogue to a close more quickly. In an ideal computer-assisted dialogue, the questions are all composed beforehand—as in a pre-written dialogue—but the questionee is faced with few questions irrelevant to her particular set of beliefs—as in an ad-hoc dialogue.

To do this, the computer will analyze information about the frequency of different beliefs, and correlations between them. In the real world, some opinions are more common than others. Nor are opinions on separate issues independently distributed:

a particular combination of beliefs may be common or uncommon. This holds even for combinations of opinions on issues that are not obviously related. For instance, among the subjects surveyed by Tamney, Johnson, and Burton [**?**] those who expressed a belief that abortion should be legally prohibited were less likely to express a belief that scientific experimentation that kills animals should be legally prohibited. Fleishman [**?**] performed a cluster analysis on political opinions in America and concluded that beliefs grouped best into six distinct clusters. One might suspect that a tendency to avoid contradiction plays a role in this clustering and that beliefs within one particular cluster tend to be consistent. In certain cases, however, even beliefs that contradict each other have been shown to cluster. Wood, Douglas, and Sutton [**?**] found that survey participants in the UK who indicated a belief that Princess Diana faked her own death to escape the public eye were *more* likely than those who did not to also indicate a belief that Princess Diana was killed in a secret plot involving the British government.

Humans have some awareness of the distribution of beliefs and use it to choose more relevant arguments. You would likely choose different arguments debating against a Democrat than against a Republican. Enabling a computer to use this information successfully in the context of Socratic dialogue depends on a successful solution to the *minimum questioning problem*: given information about the distribution of beliefs, what order should the questions of a dialogue be selected in order to lead a questionee into a contradiction asking as few unnecessary questions as possible? I devote the rest of the paper to solving this problem.

Chapter 2

PROBLEM

This chapter develops a formalization of the minimum questioning problem and proves that it NP-hard.

2.1 Configuration

A dialogue consists of a set A of *answers*, an ordered set of *questions* Q, containing disjoint subsets of A, and an ordered set of *contradictions* C, containing subsets of A such that no contradiction contains two or more answers from the same question. This restriction anticipates that the questionee will be allowed to choose only one answer to each question. Although one could imagine constructing dialogues without this restriction, or without the requirement that questions be disjoint, I have these restrictions for simplicity.

When you ask somebody a multiple choice question about their views, they are not likely to pick randomly. They will pick whichever answer they believe to be correct. To embody this I define a *belief set* as a set containing one element from each subset in Q. A belief set represents the views of one particular potential questionee. A singular element in a belief set is a *belief*, and it represents the answer the questionee would pick if asked the question to which the belief is an answer choice.

An effective computer questioner should also incorporate knowledge about the covariances between beliefs and between combinations of beliefs. For this purpose, I define the *population* P, a set of belief sets. The belief sets contained in the population represent the opinions and combinations of opinions that the questioner is likely to encounter. In other words, it is like the questioner conducted an opinion survey among potential questionees consisting of every question in the dialogue. The population

$A = \{1, 2, 3, 4, 5, 6, 7\}$ $Q = \{Q_1, Q_2, Q_3\}$
$Q_1 = \{1, 2, 3\}$ $Q_2 = \{4, 5\}$ $Q_3 = \{6, 7\}$
$C = \big\{\{1, 6\}, \{2, 7\}, \{2, 4, 6\}\big\}$
$P = \big\{$
$\{\mathbf{1}, 4, \mathbf{6}\}$,
$\{\mathbf{2}, 5, \mathbf{7}\}$,
$\{\mathbf{2}, \mathbf{4}, \mathbf{6}\}$
$\{3, 5, 6\}$
$\{3, 5, 7\}$
$\big\}$

Figure 2.1: An instance of the minimum questioning problem

describes the results. The population determines an empirical joint distribution of the potential questionees' beliefs, and by counting the proportions of beliefs sets that contain a given belief, the questioner may determine the conditional probability that a questionee whose belief set is unknown holds any particular belief, given that she holds any other combination of beliefs.

All the structures in the setup of the minimum questioning problem are illustrated in figure 2.1. There are seven answers, broken into one question with three answers, and two questions with two answers. There are three contradictions with two questions each. The population is specified, and contains 5 belief sets. The first three belief sets listed contain contradictions, which are highlighted with boldface text. The last two do not. From the population, we can determine, for example, that the probability that a questionee believes answer 1 is 1/5, since there are 5 belief sets in the population, and only the first belief set listed contains the belief 1. But the conditional probability that the questionee believes 1, given that the questionee believes 4 and 6, is 1/2, since only two entries in the population (the first and third listed) contain 4 and 6, and one of those contains 1.

2.2 Execution

2.2.1 Process

In an instance of the problem, a dialogue and population are specified. One belief set is then randomly selected from P to be the *questionee's belief set*. The

questioner does not initially know the specific contents of the selected belief set, but he does know the population from which the belief set was selected. This process is designed to embody how a questioner in reality might not have detailed knowledge about the beliefs of a particular questionee, but would have a general sense of what a person is likely to believe.

Play proceeds in rounds. In each round, the questioner asks a question that has not already been asked. The element of the selected belief set corresponding to that question is then revealed to the questioner. Using this new knowledge, he may update the probabilities he attaches to each belief being contained in the questionee's belief set. For instance, in Figure 2.1, if the questioner first asked Q_2, and the response was 5, he would remove from consideration any belief sets in the population that contained an answer to Q_2 other than 5, resulting in the following conditional population.

$$P' = \{ \quad \{\mathbf{2}, 5, \mathbf{7}\}, \\ \{3, 5, 6\}, \\ \{3, 5, 7\}\}$$

2.2.2 Termination

Play may terminate in two ways:

1. **Contradictory termination.** Play terminates when the questioner has revealed a contradiction. That is, there exists a contradiction such that every answer in the contradiction is in the questionee's belief set, and has been revealed by the questioner.

2. **Non-contradictory termination.** Play terminates when the questioner has ruled out every contradiction. This means the questioner has enough information to determine that the questioner does not believe any of the contradictions in the dialogue. This occurs when the union of all the questions that have been asked is a hitting set of C, but no element of C is a subset of this union. That is, the union of the full sets of answers to all the questions that have been asked contains at least one element of every contradiction, but no entire contradiction.

The goal of the questioning algorithm is to select the question at each round that minimizes the number of rounds before termination. Intuitively, it should be more difficult to terminate the dialogue in the non-contradictory case. A questioner must only be right about part of a questionee's belief set in order to demonstrate a contradiction–but demonstrating non-contradiction involves almost the entire belief set.

2.3 NP-Hardness of Minimum Questioning Problem

Theorem 1. *The minimum questioning problem is NP-hard.*

Proof. I shall show a reduction from the set covering problem, which is known to be NP-complete [**?**].

Begin with a universe U and a set S containing subsets of U, such that $\bigcup_{S_i \in S} S_i = U$. A set covering is a set $C \subset S$ such that $\bigcup_{K_i in K} K_i = U$. The set covering problem is to find the set covering of minimum size.

Now reduce the set covering problem into an instance of the minimum questioning problem. There will be one contradiction C_i corresponding to each element u_i of U, and one question Q_i corresponding to each subset S_i in S. There will be a population P consisting of only one belief set B.

For each subset S_i in S, do the following:

1. Create one answer for each element $s \in S_i$. Put this answer in Q_i. Since s is an element of U, there is a contradiction that corresponds to s. Insert the answer into that contradiction.

2. Create one additional answer. Put this answer in Q_i but not in any contradiction. Also put this answer in B. This is the answer of Q_i that the questionee believes. Thus, when the questioner asks Q_i, it will rule out all the contradictions containing the other answers to the question, which correspond to a certain set (S_i) of elements in U. Thus, as S_i "covers" part of U, Q_i "rules out" a subset of C.

Now a valid instance of the minimum questioning problem has been constructed: by construction, every answer is contained by one and only one question, and no

contradiction contains two elements contained by the same question, since no element appears twice in the same $S_i \in S$.

This instance of the minimum questioning problem may only be terminated through non-contradictory termination, since no belief set in the population contains a contradiction. Thus, a solution to the minimum questioning problem gives a set $Q' \subset Q$ such that the answers in B given to the questions in Q' "rule out" each contradiction in Q, and Q' will be of minimum size. Q' corresponds to a set $C \subset S$, that will be a set covering of U of minimum size. Thus, since constructing and solving an instance of the minimum questioning problem yields a solution to the NP-complete set covering problem, the minimum questioning problem in NP-hard. □

Chapter 3

APPROACHES

Since the minimum questioning problem is NP-hard, a brute-force solution is infeasible for large dialogues. Instead, I explore a greedy approach.

In this chapter, I begin by analyzing the minimum questioning problem in terms of a trade-off between exploration and exploitation. Each round, the questioner has some knowledge about the unasked questions–he can analyze the population and determine that some questions have higher probability of leading toward termination than others. He can choose to *exploit* this knowledge, and ask such a question. But he can also choose to *explore* and choose a question based on its ability to yield more knowledge, which will allow him to select better questions in future rounds. Other problems have been analyzed in these terms, particularly the multi-armed bandit problem and its variants, where the objective is to maximize payout from k slot machines whose probability of payout is initially unknown [**?**].

Next, I analyze a trade-off between pursuing contradictory termination and pursuing non-contradictory termination, and finally I develop a multi-objective utility function taking these trade-offs into account.

3.1 Contradictory Termination

3.1.1 Exploitation

Contradictory termination occurs when all components of a believed contradiction are revealed. An intermediate step towards termination, then, is the revelation of a single component of a believed contradiction. This gives rise to a natural strategy for selecting which question to ask: favor questions which are likely to reveal a component of a believed contradiction.

Consider the *exploiting Socrates*, a greedy algorithm based on this strategy. At each step, exploiting Socrates picks the question Q' which has the highest probability of revealing a component of a contradiction contained in the questionee's belief set B. This probability is given by

$$X(Q_i) = \sum_{a \in Q_i} \left[\Pr(a \in B) \sum_{C' \in C^a} \big(\Pr(C' \subset B) \big) \right] \tag{3.1}$$

where C^a denotes the set of contradictions which contain answer a. This value can be readily computed by iterating through all the belief sets in the population to determine the correct proportions. For example, in Figure 3.1 to calculate $X(Q_1)$ I consider column 1 of P. In all three belief sets, the entry in column one is given in boldface, indicating that it is a component of a believed contradiction. Thus, $X(Q_1) = 3/3 = 1$. Whereas, only two belief sets contain a bold-faced entry in the second column. Therefore $X(Q_2) = 2/3$.

This algorithm is effective at exploiting existing knowledge about the questionee's belief set to achieve progress towards contradictory termination. It performs well in situations where the trade-off between exploitation and exploration does not arise, and when it is likely that the questionee believes a contradiction. Figure 3.1 and 3.2 depict dialogues in which this is the case. In Figure 3.1, there is no difference between how well each question explores, and in Figure 3.2 the question that exploits best coincides with the question that explores best. A precise explanation of what is meant by *exploration* follows in the next subsection.

3.2 Exploration

Since exploitation is less likely to be effective without information regarding the questionee's belief set, sometimes it is a better strategy to choose a question that is less likely to lead to immediate progress, but more likely to enable the selection of more effective moves in future rounds. A measure which embodies this sort of exploration is *information gain*, which is defined as a decrease in Shannon entropy.

Shannon entropy H is formally defined as

$A = \{1, 2, 3, 4, 5, 6\}$
$Q = \{Q_1, Q_2, Q_3\}$
$Q_1 = \{1, 2\}$ $Q_2 = \{3, 4\}$ $Q_3 = \{5, 6\}$
$C = \big\{\{1, 5\}, \{2, 3\}, \{2, 6\}\big\}$
$P = \big\{$
$\{\mathbf{1}, 3, \mathbf{5}\},$
$\{\mathbf{2}, \mathbf{3}, 6\},$
$\{\mathbf{2}, 4, \mathbf{6}\}\big\}$

Figure 3.1: Each question explores equally. But Q_1 exploits best and is the optimal initial question.

$A = \{1, 2, 3, 4, 5, 6, 7\}$
$Q = \{Q_1, Q_2, Q_3\}$
$Q_1 = \{1, 2, 3\}$ $Q_2 = \{4, 5\}$ $Q_3 = \{6, 7\}$
$C = \big\{\{3, 5\}, \{2, 7\}, \{1, 4\}\big\}$
$P = \big\{$
$\{\mathbf{1}, \mathbf{4}, 6\},$
$\{\mathbf{2}, 4, \mathbf{7}\},$
$\{\mathbf{3}, \mathbf{5}, 7\}\big\}$

Figure 3.2: Q_1 explores best and exploits best. It is the optimal initial question.

$$H(P) = -\sum_{x \in X} p(x) \log_2 p(x)$$

where P is the population under consideration, X is the set of contradictions contained by any belief sets in that population, and $p(x)$ is the proportion of the number of beliefs with contradiction x compared to the total number of beliefs in X.

Entropy measures how uncertain the questioner is about the questionee's belief set. For instance, if there is only one contradiction believed by anybody in the population, and 100% of everybody in the population believes that contradiction, then $H(P) = 0$. The questioner is completely certain about the questionee's belief set. On the other hand, if there are four possible contradictions, and 25% of people in the population believe each contradiction, then $H(P) = 2$, and there is some uncertainty.

After each round, the population is reduced into another, smaller population based on the answer the questionee gives. Any belief sets in the initial population which contained an answer to the question other than that which was given will be

$A = \{1, \ldots, 15\}$
$Q = \{Q_0, Q_1, \ldots, Q_6\}$
$Q_0 = \{1, 2, 3, 4, 5\}$ $Q_1 = \{6, 7\}$ $Q_2 = \{8, 9\}$ $Q_3 = \{10, 11\}$ $Q_4 = \{12, 13\}$ $Q_5 = \{14, 15\}$
$C = \big\{\{7\}, \{9\}, \{11\}, \{13\}, \{15\}\big\}$
$P = \big\{$
$\{1, \mathbf{7}, 8, 10, 12, 14\}$,
$\{2, 6, \mathbf{9}, 10, 12, 14\}$,
$\{3, 6, 8, \mathbf{11}, 12, 14\}$,
$\{4, 6, 8, 10, \mathbf{13}, 14\}$,
$\{5, 6, 8, 10, 12, \mathbf{15}\}\big\}$

Figure 3.3: Q_0 has zero probability of revealing a component of a believed contradiction, yet, the optimal strategy is to ask Q_0 first, and then the question indexed by the answer to Q_0, since Q_0 has exploration value.

eliminated. It is possible that one answer to a question results in a new population with low entropy, whereas a different answer results in a new population with high entropy. The appropriate measure for exploration is the *information gain*, which is current entropy minus the sum of the entropies of the possible resulting populations, weighted by the probability their respective answer is given. Put formally,

$$IG(Q') = H(P) - \sum_{T' \in T} p(T')H(t)$$

where T is the set of possible new populations which will result after asking question Q' (depending on which answer is given), and $p(T')$ is the proportion of beliefs in T' compared to the proportion of beliefs in P. Refer to the greedy algorithm that, in each round, chooses that question which leads to the highest information gain as *information-gaining Socrates*.

The usefulness of this measure is illustrated in Figure 3.3. In that example, asking Q_0 has zero probability of revealing a component of a believed contradiction–but after asking Q_0 the questioner becomes completely certain about which contradiction the questionee believes in the next round. Therefore, asking Q_0 allows the questioner to terminate the dialogue more quickly on average. reduces the entropy of the population to zero.

The example given in Figure 3.4 is in the same spirit as that in 3.3. Again, Q_0 has

$A = \{1, \ldots, 10\}$
$Q = \{Q_0, Q_1, \ldots, Q_6\}$
$Q_0 = \{1, 2, 3, 4\}$ $Q_1 = \{5, 6\}$ $Q_2 = \{7, 8\}$ $Q_3 = \{9, 10\}$
$C = \big\{\{5\}, \{7\}, \{9\}\big\}$
$P = \big\{$
$\{1, \mathbf{6}, 7, 9\}$,
$\{2, 5, \mathbf{8}, 9\}$,
$\{3, 5, 7, \mathbf{10}\}$,
$\{4, 5, 7, \mathbf{10}\}\big\}$

Figure 3.4: Q_0 has the greatest exploration value, yet is not the optimal initial question because it is not enough to outweigh the exploitation value of Q_3

zero probability of revealing of a component of a believed contradiction, and reduces the entropy in the next round to 0. The difference, though, is that Q_0 has less exploration advantage over the other questions, because the initial entropy is lower. Because of this, choosing to ask Q_0 is not the best strategy. This example illustrates that neither exploration alone nor exploitation alone can be successful in all circumstances. An effective approach must incorporate both, and weigh the advantages of each against each other, taking into account the circumstance.

3.3 Non-contradictory Termination

Exploiting Socrates and information-gaining Socrates incorporated a notion of progress towards only contradictory termination. However, the condition for contradictory termination cannot be satisfied in instances where the questionee's belief set contains no contradiction. These instances require an alternative definition of progress.

The non-contradictory termination is the negation of the contradictory termination condition. To achieve contradictory termination, the questioner must uncover *one* contradiction, where *each* component is *believed*, whereas to achieve non-contradictory termination he must uncover, *for each* contradiction, *one* component which is *not* believed.

In Theorem 1 I showed a reduction between from the set-covering problem to certain non-contradictory instances of the minimum questioning problem. The general non-contradictory case bears some resemblance to the set-covering problem. In the

set-covering problem, the set of yet-uncovered elements covered by a set is known with certainty. However, the set of yet-uneliminated contradictions which asking a question will eliminate is not. The questioner knows only probabilities.

In the standard set covering problem, there is a well-known approximation algorithm for selected a set cover of near-minimum size: to successively choose the subset which covers the highest number of elements which have not already been covered. This process is known to find a set cover which is within $\log n$ of the true minimum size [**?**].

I adapt this solution, instead considering the *expected* number of yet uneliminated contradictions which will be eliminated by asking each question, since the true number is uncertain. This number is given by

$$Y(Q_i) = \sum q \in Q_i \Pr(q \notin B) * |\{C' \in C : q \in C'\}|$$

This definition assumes that C has been updated to include no contradictions that have been eliminated from possibility by answers given in previous rounds.

This strategy is somewhat at odds with the strategies developed in Section 3.1.1. Those strategies reveal beliefs which are components of many contradictions, since this maximizes the likelihood that one of those contradictions will be believed, but this approach requires the revelation of beliefs which are components of *few* contradictions. If the revealed belief is a component of many contradictions, then a different question must be asked to eliminate those contradictions.

Knowledge is still helpful in non-contradictory instances: the more information the questioner has about the beliefs held by the questionee, the more certain he can be about which questions eliminate the most contradictions. However, a useful definition of knowledge is not as readily available in the non-contradictory case. Calculating entropy requires a concept of categories. In section 3.1.1, it was feasible to categorize belief sets according to the contradiction they contained, because there are a small number of contradictions. Belief sets which contained no contradiction were grouped in a single category of their own–thus, using this definition of entropy is not as helpful in

the non-contradictory case. The number of possible ways to rule out all contradictions becomes very large as the number of questions and contradictions grows, so a similar definition of entropy for non-contradictory endpoints is infeasible. I set aside for future work the question of exploration in the non-contradictory case.

3.4 Synthesis into Multi-Objective Utility Function

In the previous sections of this chapter, I developed three distinct strategies appropriate for different instances of the minimum questioning problem. The challenge is to develop an algorithm to appropriately manage the trade-offs among the three strategies, adapting to circumstance. I present here several multi-objective utility functions which incorporate the three in various ways.

Let $\left|C\right|$ indicate the number of uneliminated contradictions, and let $\alpha(P)$ be the probability that the questionee's belief set contains a contradiction. The computer can calculate $\alpha(P)$ by counting the number of uneliminated beliefs in the population which contain a contradiction and dividing by $\left|\mathrm{C}\right|$.

For reasons outlined in the discussion in the first three sections, an effective utility function will fulfill the following properties:

1. If $X(Q_i)$, the probability of revealing a component of a believed contradiction is near 1, that question will be nearly always be picked. Since the questions will have to be picked eventually, there is no disadvantage to picking it immediately. As $X(Q_i)$ approaches 1, so does the probability it will be picked.

2. When $Y(Q_i)$, the expected number of remaining uneliminated contradictions asking a question will reveal, is the number of remaining contradictions $|C|$, that question will be picked. As $Y(Q_i)$ approaches $|C|$, the probability it will be picked must approach 1.

3. The more likely it is that that the questionee's belief set contains a contradiction, the more the utility function will favor questions that have a high probability of revealing a component of a believed contradiction.

4. The less likely it is that the questionee's belief set contains no contradiction, the more the utility function will favor questions that are likely to eliminate a large number of uneliminated contradictions.

5. The utility function favors more that reduce entropy. However, this concern matters more in the case where contradictory termination is more likely.

To fulfill requirements 1 and 2, the utility function ought to possess an upper bound (1, for simplicity) that it will assume in either of these cases to ensure a question which meets one of those conditions is picked. Toward this end, I normalize each of the three heuristics so that they do not exceed 1. $X(Q_i)$ does not need to be normalized since it already cannot exceed 1. $IG(Q_i)$ is bounded above by $H(P)$ (you can never eliminate more entropy than exists, so I use instead the quotient $\frac{IG(Q_i)}{H(P)}$, which is the proportion of entropy eliminated by asking question Q_i. Similarly, since $Y(Q_i)$ is bounded above by $\left|C\right|$, I use the quotient $\frac{Y(Q_i)}{\left|C\right|}$.

Upon these principles, I have constructed the utility function

$$U(Q_i) = \alpha(P) \cdot \left[X(Q_i) + \big(1 - X(Q_i)\big) \cdot \frac{IG(Q_i)}{H(P)}\right] + \big(1 - \alpha(P)\big) \cdot \left[\frac{Y(Q_i)}{\left|C\right|}\right]. \quad (3.2)$$

This function is a linear combination of an expression involving both X (exploitation) and the normalized IG (exploration), with the normalized Y (progress towards non-contradictory termination), weighted by α (the probability of contradiction). For high values of X, the expression contains less of IG, and vice versa.

There are infinitely many functions which fulfill the properties outlined in this section, and the provided function is only one of them. I find it an attractive candidate because I believe it is one of the simpliest, and as shall be shown in the evaluation section, the components can be analyzed separately.

Chapter 4

EVALUATION

A thorough evaluation of my preferred approach requires the systematic analysis of its performance on large dialogues and populations on a scale resembling what one might expect of human-written dialogues. Here I describe my technique for generating simulations, report and analyze my findings.

4.1 Data Generation

4.1.1 Dialogue Generation

A dialogue refers to the set of questions, answers, and contradictions independently of anybody's belief sets. I consider 4 parameters in generating dialogues.

1. Number of questions $|Q|$
2. Number of answers per question $\frac{|A|}{|Q|}$
3. Number of answers per contradictions $\frac{|A|}{|C|}$
4. Contradiction length ℓ (the number of answers in a contradiction)

First I construct Q and A, which are completely determined given $|Q|$ and $\frac{|A|}{|Q|}$. Then I find $|C|$ which is given by $|A|$ and $\frac{|A|}{|C|}$. Then, I construct C by selecting ℓ questions at random from Q, and randomly selecting one answer from each include in a contradiction. Then I add that contradiction to C and repeat this process $|C|$ times.

4.1.2 Population Generation

I generate a population by generating belief sets randomly, but not uniformly. To generate belief sets uniformly would defeat the purpose of having the population. The population is intended to resemble real-world conditions. In the real world, individuals

do not select each of their beliefs randomly. Generally, people's beliefs tend to follow patterns and have structure, with some noise. For instance, of those people who believe cocaine should not be prohibited, probably less than 50% believe that marijuana should be prohibited. However, if beliefs were chosen uniformly, there would be no such correlations. Furthermore, in a population without correlations between beliefs, there would be far less information available for the heuristics developed in the previous chapter to exploit and explore.

I choose a method of generation that introduces a structure in to the population, so that the belief sets are not uniformly distributed. My method takes three parameters:

1. The size $|P|$ of the population.
2. The number n of "base beliefs".
3. A mutation probability p.

First, I generate the n base beliefs by picking at random one answer from each question in Q, and insert the belief into P. Then, I randomly select one of the beliefs in P, and with probability p I randomly switch one of its component answers to a different randomly selected answer in its corresponding question. I repeat the previous step $|P| - n$ times.

4.2 Algorithms

I compare the performances of 7 different algorithms. The first of these is *Random Socrates*. Random Socrates, in each round, randomly selects an unasked question to ask. If an algorithm does not have better performance than Random Socrates, then the algorithm is not useful.

The next algorithm is *Peeking Socrates*. Peeking Socrates "cheats," and solves an easier version of the minimum questioning problem, where the algorithm is given certain knowledge of the questionee's belief set. This means Peeking Socrates can always terminate a dialogue in ℓ questions when the questionee's belief set is contradictory. This is the lower bound on how well any algorithm could possibly perform. In the non-contradictory case, this easier problem is still NP-hard. Therefore, the Peeking Socrates algorithm cannot both be efficient and provide a tight lower bound. However, it utilizes the well-known greedy strategy for set coverings—it picks at each step the question which rules out the most uneliminated contradictions, which will be very close to a lower bound. Whereas Random Socrates performs about the worst that any algorithm could for solving the minimum questioning problem, Peeking Socrates performs better than any algorithm should be able to.

My primary candidate for solving the minimum questioning problem I call *Synthesis Socrates*. Synthesis Socrates is the greedy algorithm which maximizes the multi-objective utility function defined in Equation 3.2.

Synthesis Socrates incorporates three objectives, X, IG, and Y. I evaluate the greedy algorithms which maximize X, IG and Y each in isolation, and the greedy algorithms which evaluate all possible combinations of 2 of these objectives, called noX, noIG, and noY. The equations for these are given below

$$noX(Q_i) = \alpha \cdot \frac{IG(Q_i)}{H}) + (1 - alpha) \cdot (Y(Q_i)/|C|)$$

This objective function incorporates the percentage of remaining entropy reduced by

asking Q_i and the expected percentage of remaining contradictions eliminated, weighting the former more when the proportion of contradictory belief sets in the remaining population is higher.

$$noIG(Q_i) = \alpha \cdot X(Q_i) + (1 - \alpha) \cdot Y(Q_i)/|C|$$

This objective function incorporates the probability of revealing a component of a believed contradiction and the expected percentage of remaining contradictions eliminated by asking Q_i, weighting the former more when the proportion of contradictory belief sets in the population is higher.

$$noY(Q_i) = X(Q_i) + (1 - X(Q_i)) \cdot \frac{IG(Q_i)}{H})$$

This objective function favors questions which have a high probability of revealing a component of a believed contradiction—but if that probability is low, favors questions which eliminate a high proportion of remaining entropy.

4.3 Hypotheses

Before proceeding to the data, I formalize several hypotheses regarding the relative performance of these algorithms.

Hypothesis 1a Random Socrates will, on average, ask a higher proportion of questions before termination than the other six algorithms.

Hypothesis 1b Peeking Socrates will, on average, ask a lower proportion.

Hypothesis 2a On experiments where the questionee's belief set contains a contradiction, a version of an algorithm without it's Y component will perform better. That is, NoY will perform better than Synthesis socrates, IG will perform better than NoX, and X will perform better than NoIG.

Hypothesis 2b On experiments where the questionee's belief set contains no contradiction, a version of an algorithm without it's X component will perform better. That is, NoX will perform better than Synthesis Socrates, IG will perform better than NoY, and Y will perform better than NoIG.

Hypothesis 3 Versions of algorithms including IG will see a slight boost in performance.

Parameter	Base Value	Min	Max	Step
$\lvert Q\rvert$	30	20	70	10
$\frac{\lvert A\rvert}{\lvert Q\rvert}$	3	1	6	1
$\frac{\lvert A\rvert}{\lvert C\rvert}$	4	1	6	1
ℓ	2	2	7	1
$\lvert P\rvert$	250	50	300	50
n	50	3	8	1
p	0.55	0.1	1	0.15

Table 4.1: Parameters used to generate this data set

Hypothesis 4 Synthesis and noIG will be not be the top performers in contradictory instances alone, or non-contradictory instances alone—but they will perform best when all cases are combined.

4.4 Data Generation

To test these hypotheses an algorithm, I randomly generate a set of experiments by specifying parameters for a base experiment and then vary one parameter at a time, leaving the other parameters at their base experiment levels.

In the data set used in the following analysis, for each set of experiment parameters, I generated 200 experiments, and ran the algorithm on each, measuring the number of questions asked until a termination condition was reached. The base parameters and range which each parameter varied inside is described in the Table 4.4.

For much of my analysis, I distinguish between the experiments which involved a questionee who believed a contradiction versus those which did not. In this data set, 5245 experiments involved contradictory belief sets, and 1002 involved belief sets which contained no contradiction. Intuitively, this proportion is mainly driven by the values of $\frac{|A|}{|Q|}$ and $\frac{|A|}{|C|}$. The more answers per question the more possible belief sets there are, but higher the proportion of contradictions to total answers, the more of these belief

sets are contradictory. This intuition could be explored in a dataset which varied these parameters, but this one does not.

4.5 Results

4.5.1 Contradictory Experiments

Figure 4.1 compares the performance of the algorithms, on the subset of experiments which were contradictory.

This figure alone contains enough information to start evaluating some hypotheses. Hypotheses 1a and 1b seem satisfied in the contradictory case. NaN, achieved by Cheating Socrates, is a substantially lower proportion than that achieved by any other algorithm, as expected. 0.521, achieved by Random Socrates, is higher than any other proportion–however it does not seem statistically distinguishable from the NaN and NaN achieved by Non-Contradictory Socrates and Exploring-Only Socrates, respectively.

Hypothesis 3 is egregiously violated in this data set. Exploiting-only Socrates seems to substantially outperform any other algorithm besides Cheating Socrates, asking on average only NaN of the questions before termination, whereas I hypothesized that both Exploring-Exploiting and Contradictory Socrates would outperform Exploiting-only Socrates, because of situations such as illustrated in Figure 3.3. This would seem to indicate that, contrary to my argument in Section **??**, including an Information Gain component is a poor strategy for a questioning algorithm, since the absence of such a component is the salient feature of the exploiting-only algorithm.

Meanwhile, these results seem to be partially compatible with Hypothesis 2a. Contradictory Socrates beats out Exploring-Exploiting Socrates, indicating that Exploring-Exploiting Socrates' caution to assume that a questionee's belief set contains a contradiction is indeed a disadvantage when the questionee's belief set does contain one. This finding would be further confirmed if a future data set contained a Contradictory Exploiting-Only Socrates, which outperformed Exploiting-only Socrates on contradictory experiments.

4.5.2 Non-Contradictory Instances

Now I turn to examining the performance of these algorithms on non-contradictory instances. The most striking feature of these results in Figure 4.2 is how they contravene Hypothesis 1a. In fact, in this non-contradictory case, Exploring-Only, Exploiting-Only, and Exploring-Exploiting Socrates perform slightly *worse* than Random Socrates, requiring approximately 5% more of the questions to terminate, on average. The smaller variance of the performance of these algorithms indicates that they do so rather consistently. This result is more interesting than if the algorithms simply failed, and performed statistically similar to Random Socrates. The curious suggestion is that an algorithm, rather than maximizing these utility functions, used those utility functions to *eliminate* choices, would perform slightly better than Random Socrates.

Non-Contradictory Socrates performs the best on average of all algorithms except for Cheating Socrates on the Non-Contradictory instances–however only by a slight amount, and with a high variance. This is compatible with Hypothesis 2b, but is far from compelling evidence in favor of it.

The performance of Contradictory Socrates is surprising. Its utility function contains nothing in common with that of Non-Contradictory Socrates which performed first best, and contains every component of the algorithms which performed worse than Random, yet it performs second best, itself.

4.5.3 General Case

I turn now in figure 4.3 to examining the entire dataset, containing both contradictory and non-contradictory instances.

The final outcome of this graph is influenced by the previously mentioned proportion of contradictory instances to non-contradictory instances (5245/1002), in turn influenced by the arbitrarily chosen values of $\frac{|A|}{|Q|}$ and $\frac{|A|}{|C|}$. Given a higher proportion of contradictory instances, it is likely the final results would be more favorable to Exploiting-only Socrates, the best performing algorithm (aside from Cheating Socrates) in the contradictory case.

Given the high proportion of non-contradictory instances, it is somewhat surprising to note that–in direct contravention to Hypothesis 2c–the algorithm which performs the best in the general case–albeit with a high variance and not by a compelling margin–is Contradictory Socrates, which performed second-best (aside from Cheating Socrates) in both the non-contradictory case and the contradictory case.

4.6 Discussion/Threats To Validity

These results call into question the relevance of the analysis in section **??**. Particularly salient is the fact that Exploring-Exploiting Socrates displayed no advantage to Exploiting-only Socrates. One possible explanation for this is that, examples such as example 3.3, are contrived, and that exploration turns out only to be useful in very rare circumstances. The question would be whether these circumstances would actually be rare among dialogues that would exist in real life, or is their rarity in this data set purely a function of my random dialogue generation strategy.

Another highlight of these results is the difficulty of the non-contradictory case. The strategy I developed specifically for this case–maximizing the expected number of additional eliminated contradictions–appears to be only a very slight improvement over Random Socrates. Unless these is a more effective solution to these cases, the best option might be to forget about the non-contradictory case altogether, where improvement cannot be made, and focus only on contradictory cases, where improvement seems possible.

Finally, I should emphasize the most encouraging result. The best-performing algorithm in the contradictory case, Exploiting-only Socrates, on average required less than half the number of questions which Random Socrates required to terminate. This shows the potential for progress on the minimum questioning problem. With more work and a better understanding of the forces underlying these results, perhaps even better results are possible.

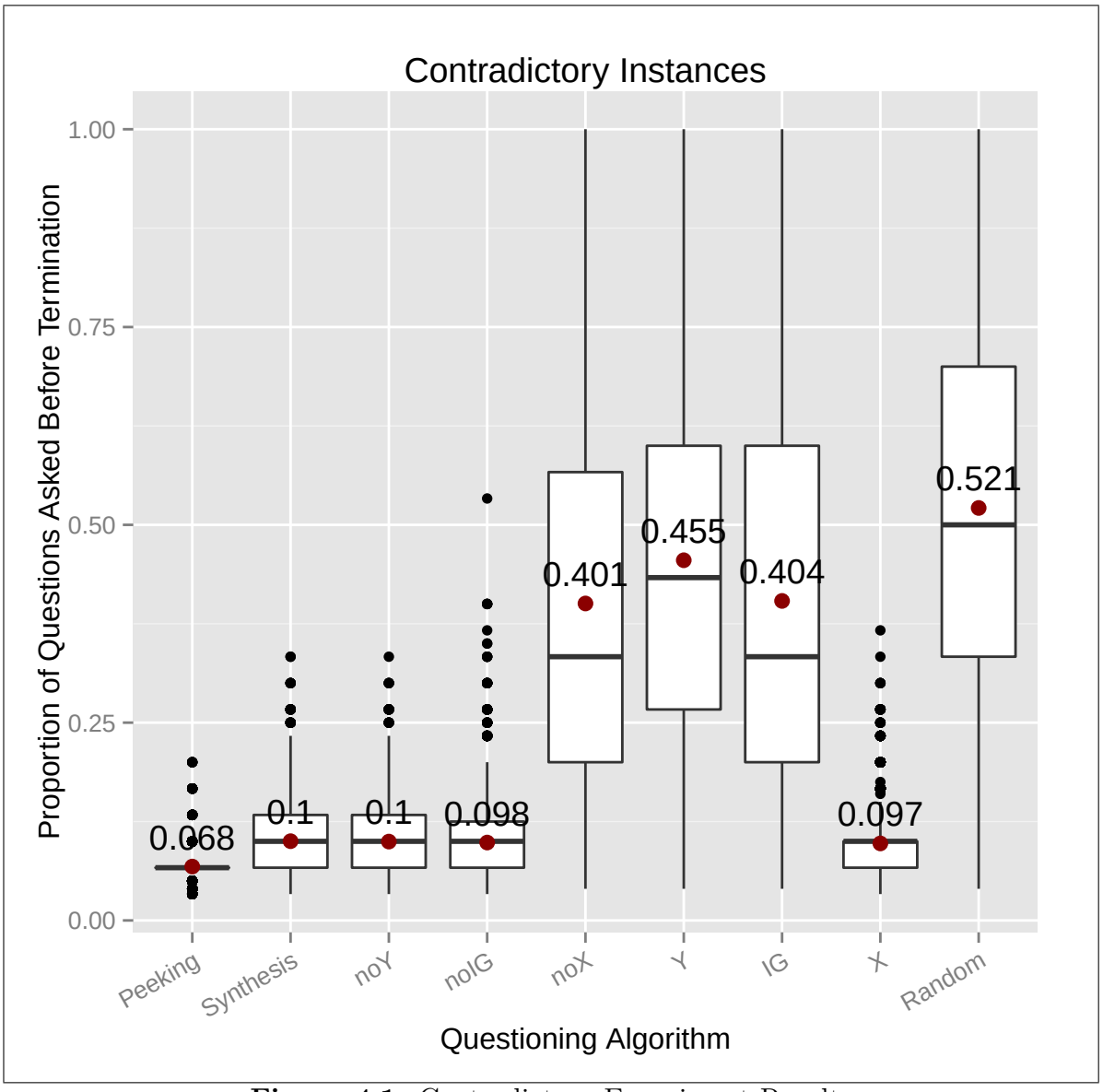

Figure 4.1: Contradictory Experiment Results

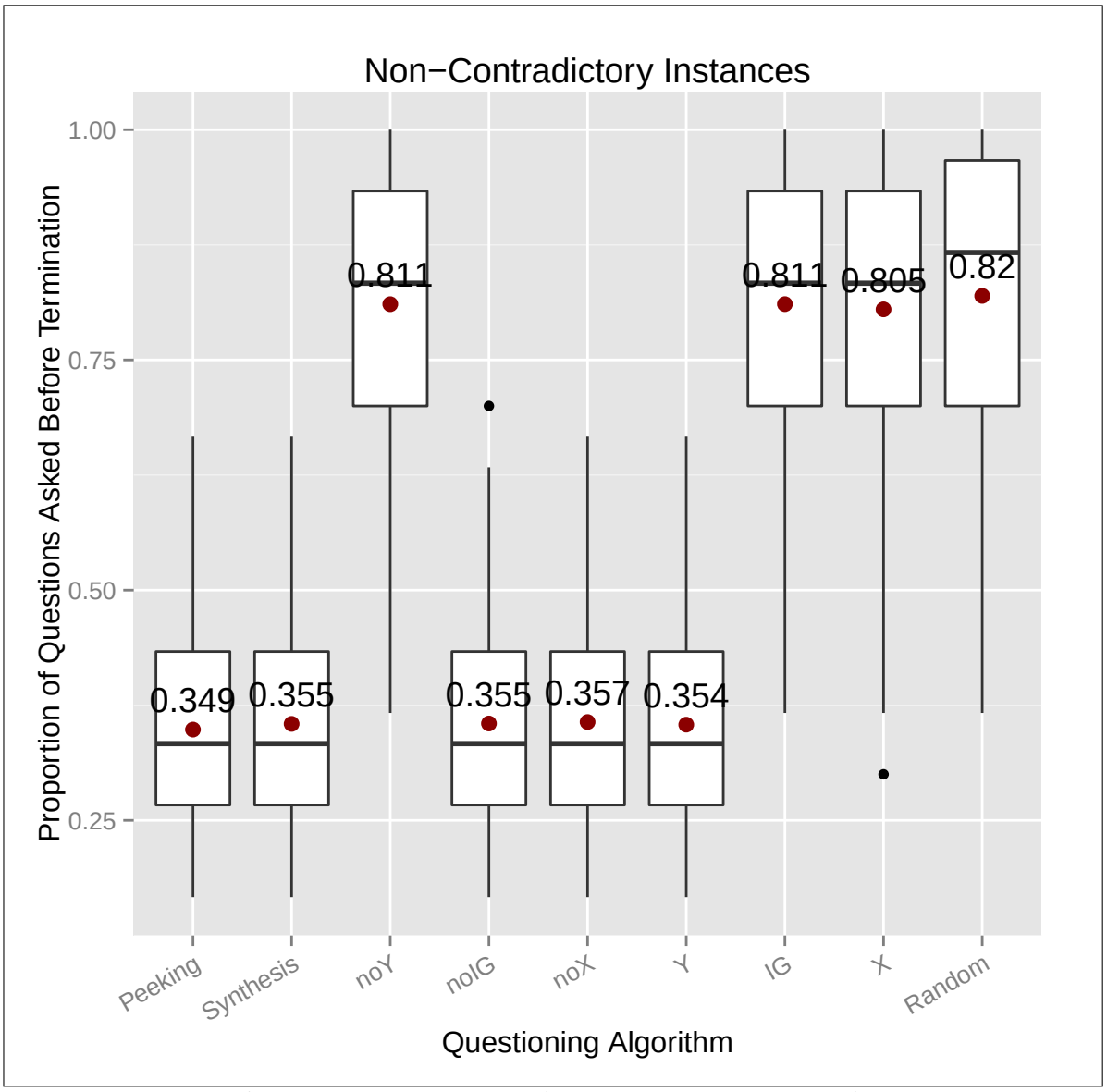

Figure 4.2: Non-contradictory Experiment Results

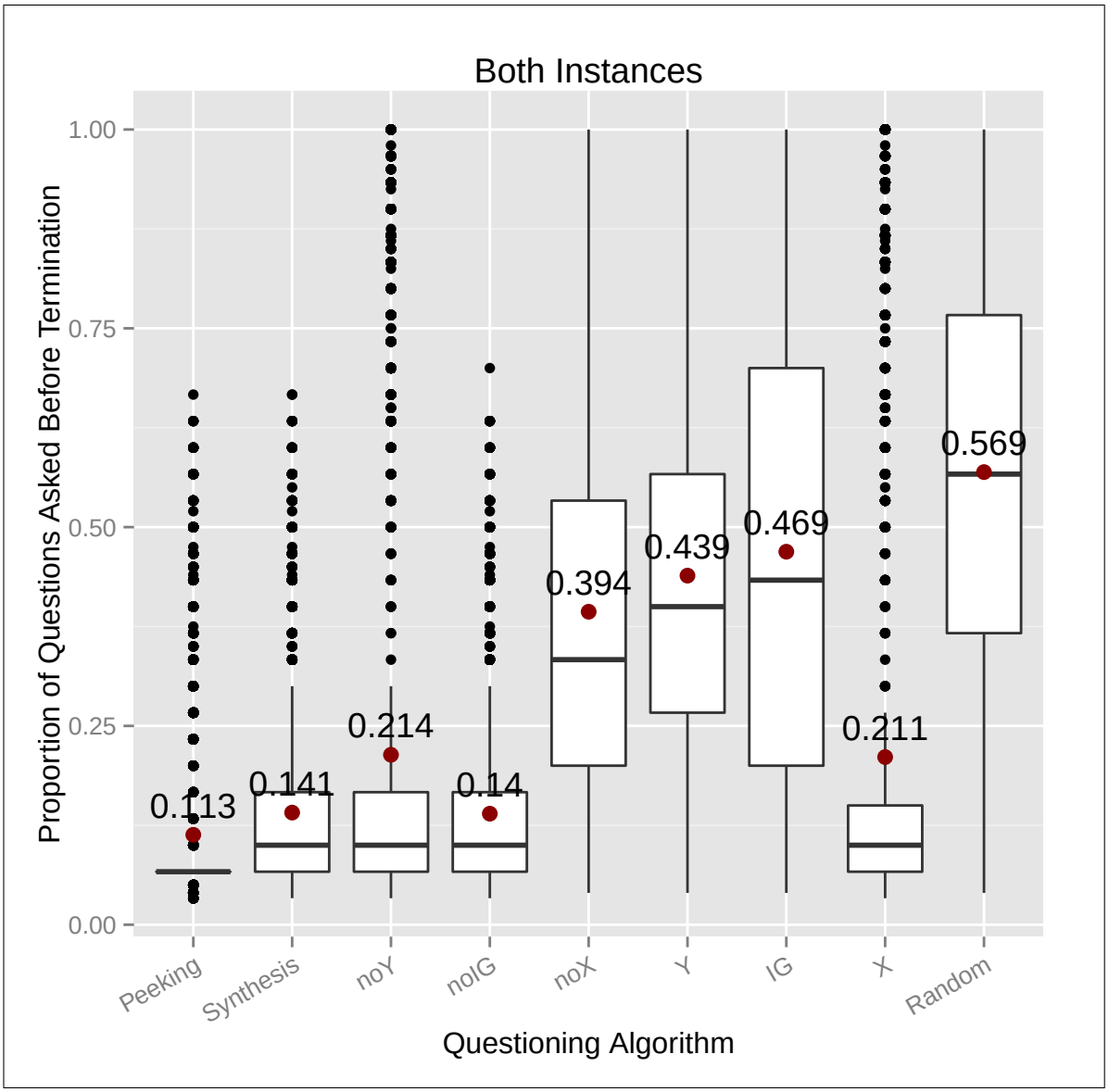

Figure 4.3: Aggregate Experiment Results

Chapter 5

RELATED WORK

This project contributes to a broad literature in artificial intelligence exploring argumentation theory and its applications. One objective of researchers in this area is the development of techniques for selecting an optimal argument. Hunter [**?**] defines a notion of the *impact* of an argument, and give an approach for identifying the argument of highest impact in an emphargument tree. Impact in their conception encompasses two ideas: *depth*, which is a measure of how long and complicated an argument is (the shorter, the better)–and *resonance,* which captures how an argument interacts with the existing beliefs of its audience. This loosely resembles my project. My criterion of optimality is the number of questions asked before termination. The number of questions part is analogous to depth, and the "before termination" part reflects audience beliefs–since termination in my framework requires the revelation or elimination of contradictions. I depart from Hunter [**?**] by treating beliefs as unknown instead of completely certain, and by choosing to model Socratic dialogue instead of a generic argumentation framework.

Other research in argumentation theory focuses on assisting humans to identify optimal arguments, rather than identifying optimal arguments themselves. Verheij [**?**] has produced "ArguMed," a software tool intended to help lawyers analyze their legal arguments to probe for weaknesses. Their research shows that structured representations of argumentation are not just interesting theoretical constructs, but they can be also be useful in practice.

My framework of modelling Socratic dialogue distinguishes my project from the bulk of other research in argumentation theory. It is common to model argumentation

in a framework of "defeasible logic," usually a variant of the Toulmin model of argumentation. Verheij [**?**] describes the history of the adoption of that model in computer science. The Toulmin model describes an *attacking* relationship between arguments. If an argument X attacks an argument Y, then Y is 'defeated' unless an argument Z has been made that attacks X and that has not been itself defeated. Researchers have used the Toulmin model to describe 'dialogue games.' A typical example is described by Jakobovits and Vermier [**?**] who describe and analyze a game where one player makes an initial argument, and then the two players alternate turns making arguments that attack their opponents' arguments. The game terminates when one of the players cannot attack the other's arguments anymore. This research investigates a dialogue game apart from Toulmin model. The Toulmin model is flexible enough to describe conventional argumentation in a number of settings because it distills whatever complicated structure an argument may have into a simple "attacks" relationship. However, Socratic dialogue does not involve participants launching attacks and counterattacks upon their opponent's arguments. In fact, only one participant, the questionee, delivers arguments at all. The questioner's moves consist not of making arguments but of eliciting argument from the questionee. Caminada [**?**] identifies Socratic dialogue as a unique form of argument, and develops a two-player model of what he terms "hang yourself (HY)" arguments, which are described in the same terms as defeasible argumentation, and have many interesting properties of interest to philosophers of argumentation. Caminada, a philosopher, notes the relevancy to the fields of computer science and artificial intelligence, but does not develop these lines of inquiry. Departing from Caminada, I dispense with the Toulmin model and defeasible argumentation frameworks in favor of a simpler one-player model designed to describe this unique situation, with applications in mind.

Chapter 6

CONTRIBUTIONS AND FUTURE WORK

In this chapter, I reflect on the contributions of my work presented in this thesis and propose directions for future work.

6.1 Contributions

The contributions of my work presented in this thesis are:

1. I identified a domain of rhetoric, socratic dialogue, which had previously remained largely unstudied by computer scientists, and outlined a potential application of computation to this domain.

2. I developed the concept of a computer-assisted socratic dialogue, and formalized the *minimum questioning problem* which must be solved in order for a computer-assisted socratic dialogue to have an advantage over classical socratic dialogue.

3. I proved that the minimum questioning problem was NP-hard.

4. I analyzed the minimum questioning problem in terms of a trade-off between exploration and exploitation, and also in terms of a trade-off between performing well when the questionee believes a contradiction and performing well when she doesn't. I developed three heuristics embodying these trade-offs, and synthesized them into one utility function.

5. I developed a system for randomly generating sample dialogues and populations to evaluate the effectiveness of approaches to the minimum questioning problem.

6. I collected preliminary results which suggest that Contribution 4 isn't actually much of a contribution, and that the trade-off between exploration and exploitation isn't very important to achieving a useful solution to the minimum questioning problem–but that demonstrate that some improvement is possible using only the concept of exploitation.

7. I designed a multi-objective greedy algorithm which was able to achieve substantial reductions in the number of questions required to terminate a dialogue when compared to random selection.

6.2 Future Work

Future work in computer-assisted socratic dialogue includes

1. The ability to handle the unknown. Right now, each questioning algorithm assumes that the questionee's belief set is contained in its population. That amounts to assuming that the questionee's belief set is something it has encountered before. But this is not guaranteed to be true in practice. The questionee's belief set might be something similar, but not identical to anything encountered before. Or it might be something else entirely. A practical questioning algorithm should be able to handle these situations gracefully.

2. Handling incomplete belief sets. Right now, the "population" structure only contains belief sets which provide answers to every question. But the only way such a structure could be collected in real life would be by asking questionees every question of the dialogue. But the point of the dialogue is to be able to ask each questionee a small number of questions–asking every question would defeat the purpose. A practical questioning algorithm would support a population containing incomplete belief sets. Either it would somehow interpolate the missing answers in an incomplete belief set, or use a different scheme of calculating the conditional probabilities involved. This would permit the questioner to be deployed with no or only a very small population, and to improve its performance on the job.

3. Further exploration on the parameter space of dialogue generation. Are the results found in this experiment only applicable to the sets of parameters chosen for this test data? Do they apply more generally? Are there sets of parameters in which *exploration* is more useful?

4. Application. Does this research adequately anticipate the challenges that would arise in a computer program actually deployed to engage in Socratic Dialogue with participants? Is the formalization of socratic dialogue presented here an effective means of communicating real-life systems of thought? Could it be modified or made more flexible to allow for more freedom in designing arguments without losing its advantages?