NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INFORMATION SELECTION IN INTELLIGENCE PROCESSING

by

Yuval Nevo

December 2011

Thesis Advisors:	Moshe Kress
	Nedialko B. Dimitrov
Second Reader:	Michael Atkinson

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 2011	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Information Selection in Intelligence Processing	5. FUNDING NUMBERS
6. AUTHOR(S) Yuval Nevo	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943–5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

In many intelligence agencies, the processing of data into usable information ready for analysis poses a significant bottleneck. Typically, much more data is available than what can be processed in the limited time available for processing.

We formulate the problem faced by an intelligence collection unit, when processing incoming raw information for delivery to intelligence analysts, as an exploration-exploitation problem: the processor has to choose between exploring for new sources of relevant information and exploiting known sources.

To address the exploration-exploitation problem, we develop a mathematical model of the processor's knowledge and examine algorithms that allow the processor to maximize the discovery of relevant data given a time limit. We derive insights on the performance of different algorithms using a simulated case study.

14. SUBJECT TERMS Exploration Exploitation problems, Intelligence Cycle, Intelligence Processing	15. NUMBER OF PAGES 135
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540–01–280–5500 Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

INFORMATION SELECTION IN INTELLIGENCE PROCESSING

Yuval Nevo
Captain, Israel Defense Forces
B.Sc., The Hebrew University of Jerusalem, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 2011

Author: Yuval Nevo

Approved by: Moshe Kress
Thesis Co-Advisor

Nedialko B. Dimitrov
Thesis Co-Advisor

Michael Atkinson
Second Reader

Robert F. Dell
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In many intelligence agencies, the processing of data into usable information ready for analysis poses a significant bottleneck. Typically, much more data is available than what can be processed in the limited time available for processing.

We formulate the problem faced by an intelligence collection unit, when processing incoming raw information for delivery to intelligence analysts, as an exploration-exploitation problem: the processor has to choose between exploring for new sources of relevant information and exploiting known sources.

To address the exploration-exploitation problem, we develop a mathematical model of the processor's knowledge and examine algorithms that allow the processor to maximize the discovery of relevant data given a time limit. We derive insights on the performance of different algorithms using a simulated case study.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

CCIRM: Collection Co-ordination and Intelligence Requirements Management

KG: Knowledge Gradient

KGEF: Knowledge Gradient Exploration First

IR: Information Retrieval

OR: Operations Research

PE: Pure Exploitation

POMDP: Partially Observable Markov Decision Process

R&S: Rating and Selection

VDBE: Value Difference Based Exploration

WEF: Wide Exploration First

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

One of the key stages in producing intelligence is *Processing and Exploitation*. Within this stage, the collected raw data is transformed into usable information. In modern intelligence agencies, one of the main obstacles in the Processing and Exploitation stage is the abundance of information, which makes differentiating between relevant and irrelevant information a difficult task. Due to time constraints, an intelligence processor of collected raw data, called henceforth a *collector*, cannot process all the collected intelligence items and therefore some screening procedure is needed. In this research we address the *information selection problem*: Which intelligence items should the collector screen in order to maximize the expected amount of relevant information screened?

The information selection problem can be seen as a part of a broader class of problems called *exploration-exploitation problems*. In an exploration-exploitation problem one has to repeatedly choose between several alternatives, and faces the tradeoff between exploring (investigating new alternatives) and exploiting (utilizing familiar alternatives). The information selection problem has unique characteristics, making it a relatively difficult exploration-exploitation problem. Specifically, intelligence sources are dependent; the information gained from the screening process of one source can be used to better estimate the relevance value of other sources.

In order to handle the information selection problem, we develop a mathematical model of the information screening process. The model handles a situation in which a collector faces a pool of intercepted conversations, which he needs to screen. We explored several selection algorithms that would allow the collector to detect as many relevant information items as possible. Based on the mathematical model, we constructed a simulation of the screening process. We then examined the performances of several selection algorithms, using a scenario based on the terrorist network behind the U.S. embassy attack in Tanzania in 2007.

The main contributions of the thesis are the mathematical model of the screening process, the selection algorithms and several important insights detailed below:

- Simple selection algorithms, which we examined, performed much better than anticipated. We anticipated that a simple greedy algorithm and another basic algorithm called “Softmax” would perform much worse than more advanced algorithms. However, the performance of these algorithms was quite well compared to the advanced algorithms. We speculate that the dependencies among the alternatives are the main reason for that performance.
- The algorithms which showed the best performance are an algorithm based on the Knowledge-Gradient policy and an intuitive heuristic for screening the conversations. The Knowledge-Gradient policy is an exploration method in which one chooses the alternative that is most likely to change its beliefs regarding the value of the different alternatives.
- The mean number of conversations between the different persons is a significant factor in the performance of the algorithms. When the mean number of conversations is small, there is no significant difference between the performances of the different algorithms.

ACKNOWLEDGMENTS

I wish to express my deep appreciation to my advisors, Moshe Kress and Nedialko Dimitrov, for their patient guidance which enabled me to complete this research, and moreover taught me many valuable lessons that I am sure would accompany me for a long time. A special gratitude to Michael Atkinson: his comments allowed me to significantly improve this thesis.

I would also like to thank the Operations Research faculty for an amazing year in which I learned a lot of valuable techniques, skills and got a first glimpse on the fascinating world of Operations Research.

THIS PAGE INTENTIONALLY LEFT BLANK

I. BACKGROUND AND PROBLEM DESCRIPTION

A. INTELLIGENCE PROCESSING

1. The Intelligence Cycle

According to the DoD dictionary, *Intelligence* is defined as[1]: "The product resulting from the collection, processing, integration, evaluation, analysis, and interpretation of available information concerning foreign nations, hostile or potentially hostile forces or elements, or areas of actual or potential operations." However, Intelligence can also be described as a process (Johnson, 2007). This process is commonly represented by the *Intelligence Cycle* (Hulnick, 2006). Although Hulnick criticizes this model, he states that "no concept is more deeply enriched in the literature [of Intelligence studies] than that of the intelligence cycle."

The intelligence cycle consists of five stages (CIA, 1999): (1) Planning and Directing/Needs; (2) Collection; (3) Processing and Exploitation; (4) Analysis and production; (5) Dissemination. First, during the planning and directing stage, the intelligence requirements of the policymakers are established. Then, at the collection stage, the raw data is gathered. Richelson (Richelson, 2007) provides a summary of different means employed to gather that data. The raw data is then converted into a usable format during the processing and exploitation stage. The data is therefore transformed into information. The information can be divided into pieces called *intelligence items*. The analysis and production stage consists of the integration and evaluation of the data, and preparation of the intelligence product. After those products are disseminated, new intelligence requirements are established, and the cycle starts again from stage (1).

In this thesis we focus on the processing stage. As mentioned above, within that stage raw data is being transformed into usable information. This is a complicated stage, and the CIA consumer's guide to intelligence (CIA, 1999) states that: "A substantial portion of U.S. intelligence resources is devoted to processing and exploitation." The transformation of the raw data might require decryption and decoding, translating the

[1] See DoD dictionary at http://www.dtic.mil/doctrine/dod_dictionary/data/i/4850.html.

data, transforming the data for computer processing, storage and retrieval, adding background information to make the data more comprehensible, etc. Some of those processes can be done automatically. However, “only the human mind can add the discernment and knowledge that makes sense of it [the information]” (Hedley, 2007).

2. Data Overload

The problem of information abundance in the modern age is described by Hedley (Hedley, 2007). A few decades ago, the main challenge for the intelligence community used to be having too little data. Nowadays, this is no longer the case. The challenge lies in “the sheer volume of information available,” as “data multiply with dizzying speed.” Therefore, “selecting and validating it [the data] loom ever larger as problems for analysts today” (Hedley, 2007).

Examples for the effect this problem has on the intelligence products can be found in the research of Gill (Gill, 2007) who claims that data overload is one of the reasons for the intelligence failures in both 9/11 and the lack of predicted weapons of mass destruction in Iraq. He states that “fundamental are the problems of overload and complexity. The very sophistication of modern information-gathering systems produces the problem of overload.” As another example, Whaley (Whaley, 1974) argues that one of the causes for the Pearl Harbor and Barbarossa strategic surprises is inability to handle large amounts of data.

Therefore, after the raw data is transformed into usable information it needs to be classified as relevant or irrelevant. This classification occurs within the processing stage (stage [3]) before the analysis stage (stage [4]). The selection of data is a complicated problem, since it requires a human involvement. Even though computerized tools that can automatically screen the data exist, these tools are not sophisticated enough to replace a human operator.

The personnel responsible for the selection of data are referred to within their organizations as analysts. However, it is important to distinguish them from the personnel who participate in the analysis and production stage (stage 4 in the intelligence cycle),

who are also referred to as analysts. We will therefore refer to the personnel performing the screening of the information as *collectors*, since the processing stage is usually conducted by data collection agencies.

3. The Information Selection Problem

Due to time constraints, the entire glut of information available cannot be screened. The collector needs to focus only on a small portion of it. However, it is difficult to accurately know in advance which portions of the available information contain relevant information. Only along the screening process, the collector can determine the probability that a certain portion of the glut of information is relevant.

The different information sources might be correlated. For example, if the collector discovers that person A has relevant information, and knows that person A is a coworker of person B, then the probability that person B also has relevant information is increased. Although this feature allows the collector to better allocate his time, this possible inference greatly complicates the problem.

Within this research, we focus on a scenario in which the collector needs to screen intelligence items from several available correlated sources. Those sources may be intercepted communication links, for example. Due to time constraints of the collector, he cannot screen all the intelligence items. We assume that the collector chooses which intelligence items to screen according to his assessment regarding the different sources, i.e., which sources are more likely to contain relevant information. The assessment of the different sources changes as the collector accumulates knowledge from the screened items and thus receives feedback regarding the relevance of the information.

Following the above discussion, the *information selection problem* is defined as follows: Which intelligence items should the collector screen in order to maximize the expected amount of relevant information screened?

B. SIMILAR PROBLEMS

In this section we review several classes of problems similar to the information selection problem. We first discuss the applicability of operations research methods for studying intelligence problems, and then we focus on some relevant classes of computer science and operations research problems.

1. Operations Research and Intelligence

The applicability of operations research (OR) methods for intelligence is reviewed by Kaplan (Kaplan, 2011), who explains what OR is, shows some key OR methods, and discusses the applicability of those methods for intelligence analysis. However, the applications proposed in his article are meant mainly for the analysis and production stage (stage [4]) in the intelligence cycle, not for the processing stage. Other applications for the analysis stage include employing OR methods for the analysis of social networks, as in (Lindelauf, 2008).

Many articles suggests employing OR methods to assist collection co-ordination and intelligence requirements management (CCIRM) (Desimone et al., 2002). Those articles discuss different methods for optimal allocation of sensors (Preece et al., 2008), as well as higher level resource allocation analysis for interdicting a nuclear program of a hostile state (Skorch, 2004).

Costica also models the selection and classification of data before delivering the information for analysis (Costica, 2010), as our research. However, Costica focuses on modeling the error rate of the screening process, and does not handle the problem of choosing what data to screen.

2. Information Retrieval

Information retrieval (IR) is defined as “finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)” (Manning et al., 2008). Although it can be applied into small scale problems such as finding a book in the library, its usual application is the retrieval of documents from a web-based storage, as is the case when

one enters a query search in Google. Many different algorithms handle this problem (Langville et al., 2005, for example), which becomes more and more important as the Internet evolves.

The IR problem and the information selection problem share some similar characteristics. First, in both problems one needs to retrieve relevant information from a large variety of possibilities. Second, both problems have a dynamic nature, as the algorithms for solving the IR problem depend on feedback regarding the relevance of the already retrieved information. Third, the sources in both problems are correlated. The correlation in the web is due to the existence of hyperlinks between web documents. Several algorithms for solving the IR problem take those hyperlinks into account (Langville et al., 2005).

However, there are substantial differences between these two problems. First, the IR problem handles data in a much larger scale, as Google, for example, searches through billions of possibilities. Second, in the information selection problem the collector receives immediate feedback on his choices that allow him to immediately adapt his assessments, unlike the IR problem.

The similarities between the problems suggest that we might attempt to employ methods used for solving the IR problem to solve the information selection problem. Although that approach might be useful, due to the differences between the two problems we decided to derive our methods from algorithms that treat other types of problems, more intuitively similar to our problem: ranking and selection problems and exploration-exploitation problems.

3. Ranking and Selection

The Ranking and Selection (R&S) problem can be regarded as “selecting the best design among a finite number of choices, where the performance of each design must be estimated with some uncertainty through stochastic sampling” (Fu et al., 2002). In other words, one is faced with several alternatives, and needs to sample them in order to determine which the best one is. Each sampling of an alternative might produce several

possible results. Specifically, for the information selection problem, an alternative can be regarded as an information source, and sampling as screening information and checking whether it is relevant or not.

Fu et al. suggest different methods to handle the ranking and selection problem (Fu et al., 2002). However, facing correlations between the different alternatives, only Frazier et al. (Frazier et al., 2010) suggest a method that takes those correlations into account. Frazier et al. state that "to our knowledge no work has been done within the R&S literature to exploit the dependence inherent in our prior belief about the value of related alternatives." We use the algorithm suggested in that article, the *knowledge gradient policy* to solve the information selection problem, as shown in Chapter III.

4. Exploration-Exploitation

Due to time and resource constraints, the collector faces a tradeoff between 1) relying on sources he is already familiar with and knows that they would produce a certain amount of relevant information, and 2) attempting to explore new sources, which might prove to be better or worse than the familiar sources. In the literature, the tradeoff between exploring (investigating new sources) and exploiting (utilizing familiar sources) is called the *exploration-exploitation* problem (Cohen et al., 2007). A common example for the exploration-exploitation problem is the multi-armed bandit problem (Berry et al., 1985). In this problem, a gambler has several levers that he can pull. Each pull returns a reward according to a distribution unknown to the gambler. The goal of the gambler is to maximize the sum of the reward accumulated from pulling the levers over time.

Although similar to the R&S problem, the objectives of the two problems are different. While the objective in the exploration-exploitation problem is to maximize the overall reward derived from choosing the different alternatives, the objective in the R&S problem is to find the best option. In a way, the R&S problem focuses only on the exploration portion of the exploration-exploitation problem, and does not take into account rewards that might be accumulated during the sampling process. Since the

objective of the collector in the information selection problem mentioned above is to get as much relevant information as possible, the problem can be regarded as an exploration-exploitation problem.

Berry et al. provide a survey of methods that address the exploration-exploitation problem (Berry et al., 1985). A useful solution for this problem is the use of Gittins Index, which offers a way to assign value for each alternative and choose the alternative with the best value (Gittins et al., 2011). Given certain assumptions, this is proven to be the optimal policy. However, these assumptions do not necessarily hold for our problem. These key assumption, which do not hold in our information selection problem are: 1) the alternatives have to be independent, the sources in our problem may not be independent; 2) infinite horizon, i.e., there is no strict time constraint, while in our problem the number of intelligence items is finite; 3) monotone decreasing value of the rewards, while in our problem the value of intelligence is not discounted. Tokic (Tokic, 2010) suggests more "flexible" algorithms, that we will use in our research.

C. CHAPTER OUTLINE

The thesis has five chapters. Following Chapter I, in Chapter II we propose a mathematical model for the information screening process. Chapter III provides several possible algorithms to handle the information selection problem. In Chapter IV we describe a simulation and a specific scenario, both used to examine the performance of the algorithms mentioned in Chapter III. Chapter V shows a comparison of the algorithms performance. In Chapter VI we summarize the research and propose possible model extensions and future research directions.

THIS PAGE INTENTIONALLY LEFT BLANK

II. THE MODEL

In this chapter we further describe the information selection problem shown in Chapter I, scoping it and stating our main assumptions. Then, we propose a mathematical model to represent this problem.

A. THE PROBLEM

1. The Communication Network

We consider a screening process where an intelligence collector (in short, collector) faces a pool of records, documenting the content of a certain communication network during a given time period. The nodes in this network may be phone numbers, e-mail addresses, fax numbers, etc. We assume that the network remains stationary throughout the screening process – no nodes are added or removed, and no new records are added to the pool.

Each record in the pool describes a conversation between two nodes. In order to get the content of the conversation, the collector needs to allocate time for screening it (listen to it, read it, etc.). We ignore the possibility of using automatic tools used for extracting information from such conversations, and assume that the collector has to go over the conversation himself.

Prior to screening a conversation, the collector only knows which two nodes participate in the conversation, without any knowledge about the content of the conversation. The collector might have some knowledge about the identity of a person behind a certain node—his names, his role in the organization, etc. The way the collector uses this knowledge is explained later on.

2. Screening a Conversation

At any given moment, the collector has access to the entire pool of records, and can screen whichever conversation that he chooses. The collector tries to determine which conversations contain relevant information. Information is relevant only if it is useful for an analyst in the Analysis and Production stage (stage four in the intelligence

cycle, as explained in Chapter I). Since we focus on the Processing stage, the question which information is considered relevant is beyond our scope of research. We therefore simply assume that the distinction between relevant and irrelevant conversations is well-defined. We assume that all the relevant conversations in the pool have the same operational value.

After screening a conversation between i and j, the collector knows for certain whether it is relevant or irrelevant (i.e., we assume there are no errors in determining the relevance of a conversation). Using Bayes theorem, the collector can then update his beliefs regarding the probability that any conversation between i and j is relevant.

3. The Relevance Value of a Node

In order to assess the probability that a conversation between two given nodes is relevant (prior to screening it) the collector can use two types of information. First, he can rely on past screenings of conversations between these two nodes, and see how many of them were relevant. Second, he can rely on the information he has about the identity of the nodes participating in the conversation. This information might include the access a person has to relevant information, his tendency to discuss such matters through the communication channel, etc. We aggregate that information into a *relevance value* assigned to each node. That relevance value is a categorical variable, indicating the likelihood that a conversation involving the node will be relevant.

After screening a conversation between i and j, the collector can use Bayes theorem to update his beliefs regarding the relevance values of nodes i and j. In addition, the collector can then update his beliefs regarding the relevance values of other nodes in the network. The collector can do that based on his assumptions regarding the connections between persons in the network. For example, a person with a relevance value x might be likely to contact persons with a relevance value y. Specifically, we assume homophily in the network (McPherson et al., 2001): Persons with high relevance value are more likely to be engaged in conversations with each other, as they might work with each other, share information with each other, etc. The homophily assumption might

not always hold. However, the model can be easily adjusted for other types of connections between the persons in the network.

Therefore, after screening a conversation, the collector can update his assessment regarding the relevance value of other nodes. Then, the collector can update his beliefs regarding the probability that a conversation between other nodes is relevant. That updating process is explained later on.

4. Direct Information on the Relevance Values

The previous section showed how the collector can infer the relevance value of a node according to the relevance of the screened conversations. However, the collector might also have direct information on the relevance values of the nodes. For example, one of the persons participating in a conversation might mention his role in the organization in which he works.

We assume that screening a conversation might result in gaining direct information regarding the relevance values of the participating nodes. Based on the gained information, the collector can update his beliefs regarding the relevance values of the nodes. For simplicity, we assume that after the collector gains such direct information, he knows with certainty the exact relevance value of a node. This assumption might seem strong, but it can be relaxed (as shown in Chapter VI). We therefore consider two situations: either a node is fully *identified* or it is *unidentified*. If the node is identified, then the relevance category is known to the collector with certainty. If the node is unidentified then the collector only knows the relevance category with probability.

Therefore, there are two possible outcomes from screening a conversation: 1) determining whether that particular conversation is relevant; 2) gaining information on the relevance value of a node participating in the conversation.

B. THE MATHEMATICAL MODEL

1. The Network

The communication network is represented by a graph. The nodes in the graph are simply the nodes of the communication network, were N represents the number of the nodes. Two nodes are connected by an edge if and only if there is at least one conversation between them in the pool. The number of conversations between two nodes is denoted by n_{ij}.

As mentioned before, each node i has a relevance value d_i which assumes a discrete set of possible values. In addition, every edge (i, j) is assigned with a parameter $p_{ij} \in [0,1]$, indicating the probability that a given conversation between i and j is relevant. The collector does not know with certainty the values of p_{ij} and might not know with certainty the values of d_i, as will be explained in the next section. We assume that p_{ij} remains constant throughout the entire screening process, and that given the value of p_{ij} the conversations between i and j are independent. Therefore, the number of relevant conversations between i and j follows a Binomial distribution with the parameters n_{ij} and p_{ij}, and each conversation k can be represented by a Bernoulli random variable, $S_{ij}^{(k)}$ whose value is 1 if the k th conversation between i and j is relevant, and 0 if it is not. In practice, the assumption that the variables $S_{ij}^{(1)}, S_{ij}^{(2)}, \ldots$ are independent given the value of P_{ij} does not always hold. For example, the conversations following a relevant conversation might be more likely to be relevant. However, we still use that assumption as it significantly simplifies the model.

2. The Collector

Since the collector has access to the entire network, he obviously knows the graph topology, i.e., the nodes and the edges, and the number of conversations n_{ij} associated with each edge. However, he doesn't necessarily know the parameters d_i, which he gradually identifies throughout the screening process, and he will never know the exact

value of any p_{ij}. He therefore has to estimate those parameters using the random variables D_i whose values are the relevance values for the nodes, and P_{ij} whose values are the probability that a conversation between i and j is relevant. D_i is therefore a discrete random variable, while P_{ij} is a continuous random variable and its values vary between 0 to 1.

There are two ways the PMF of the D_i s is updated:

1. "Sudden revelation" following a screening of a conversation – one or two nodes participating in this conversations become identified, and their PMF becomes a deterministic distribution. The PMF of the rest of the nodes is updated according to the conditional probabilities $\Pr(D_i \mid D_j)$, which are assumed to be fixed and known. This option represents a situation in which the content of the conversation provides specific information which enables the collector to determine exactly the relevance value. A node can only become identified after a sudden revelation occurs.
2. "Regular update" according the relevance of the conversation. After determining the relevance of the conversations, the PMFs of unidentified nodes in the graph are updated using Bayes rule.

As mentioned before, the collector might have some knowledge about the relevance variables D_i, derived from the content of the conversations. Even though in real life a collector might gradually gather information about a node, we assume that all the information is gathered in one instance, in one conversation: the relevance value of a node is either identified or unidentified. The random variable D_i is identified if and only if exists a value d_i such that $\Pr(D_i = d_i) = 1$. When the D_i is unidentified, the collector has no direct information about its relevance value, and he can only assess its distribution according to the values or distributions of its neighbors. When a node becomes identified, its distribution collapses into one value. The way in which a relevance variable is identified would be described in the next section.

We assume that every P_{ij} is dependent on D_i and D_j, since the relevance values indicate how likely are the nodes to be involved in a relevant conversation. We also assume that P_{ij} is conditionally independent of all the other random variables (except, of course, $S_{ij}^{(k)}$) given D_i and D_j. The conditional probability $\Pr(P_{ij} = t \mid D_i = d_i, D_j = d_j)$ can be derived, for example, from statistics over previous screening processes.

While the different random variables D_i are dependent, we assume that they satisfy the Markov Property, i.e., given the relevance values of the nodes adjacent to i, D_i is independent of all other D_j in the network. We assume that the collector has a prior joint distribution over the set of all the unidentified D_i, and therefore, in particular, he knows the conditional probabilities between any two D_i and D_j. That prior joint distribution is updated throughout the screening process.

3. Screening the Conversations

The collector has resources constraints represented by T, an integer indicating the total number of conversations the collector can screen. Clearly $\sum_{(i,j)} n_{ij} > T$, i.e., the collector cannot screen all the conversations. We assume that screening each conversation requires the same amount of resources.

As stated before, screening a conversation between i and j can result in one or two of the following outcomes: 1) Determining that the conversation is either relevant or irrelevant; 2) The variable D_i or D_j involved in the conversation is identified. We assume the collector has no errors.

If both relevance values are already identified, the probability that a conversation between i and j is relevant is simply $\Pr(S_{ij} = 1) = p_{ij}$. Since the collector doesn't know p_{ij}, he can estimate it using the equation:

$$\Pr(S_{ij}^{(k)} = 1) = \int_{t=0}^{1} \Pr(P_{ij} = p) \Pr(S_{ij}^{(k)} = 1 \mid P_{ij} = p) dp = \int_{t=0}^{1} \Pr(P_{ij} = p)\, p\, dp = \mathrm{E}(P_{ij}) \qquad (2.1)$$

where the distribution of P_{ij} depends on the relevance values d_i and d_j and the number of relevant and irrelevant conversations between i and j which have already been screened.

After screening a conversation, the collector can update the distribution of P_{ij} using Bayes theorem, as will be explained in the next section. If at least one of the nodes i and j is unidentified, there is a probability that an unidentified relevance value will be identified, denoted by c. We assume that c is independent of whether the conversation is relevant or not. The value of c might depend on the relevance value of the node (d_i) but for simplicity we assume that the value of c is the same for all relevance values. We assume that c remains constant throughout the screening process. If both nodes i and j are unidentified, we assume that each one of them is identified with probability c independently. We assume that the value of c is known to the collector, as it can be easily deduced from statistics on other screening processes that already took place. The probability to identify a node is independent of the probability that the conversation is relevant.

The collector tries to find a policy for choosing which conversation to screen at each iteration, in order to maximize R, the number of identified relevant conversations. Other possible goals, such as identifying as many relevance values as possible, are beyond the scope of this research.

4. Model Parameters - Summary

Symbol	Type	Meaning
N	Parameter	Number of nodes in the graph
n_{ij}	Parameter	Number of conversations between i and j
d_i	Parameter	Relevance value of i
p_{ij}	Parameter	Probability that a conversation between i and j is relevant
D_i	Variable	A random variable used to estimate the relevance value of i
P_{ij}	Variable	A random variable used to estimate the probability that a conversation between i and j is relevant
$S_{ij}^{(k)}$	Variable	The relevance of the k th conversation between i and j
T	Parameter	The maximal number of conversations the collector can screen
c	Parameter	The probability that an unidentified relevance value is identified
R	Variable	The number of relevant conversations the collector has identified

Table 1. Summary of the model parameters

5. Model Assumptions – Summary

- Two nodes in the network are considered connected if and only if there is at least one conversation between them ($n_{ij} \geq 1$).
- The collector has access to any conversation in the network.
- The relevance of a node can be represented by the categorical value d_i.
- The different p_{ij} and d_i remain constant throughout the screening process.
- The conditional probability $\Pr(P_{ij} = p_{ij} \mid D_i = d_i, D_j = d_j)$ is the same for all (i, j) and is known to the collector.

- Every P_{ij} is conditionally independent of all the other variables (except $S_{ij}^{(k)}$) given the values of D_i, D_j. Every D_i is independent of the other relevance values given the values of its neighbors.
- The collector has a prior distribution over all the D_i in the graph.
- All the conversations in the graph have the same value, and require the same amount of resources to screen.
- The relevance values of the nodes are either identified or unidentified. An unidentified node can only be identified if the collector listens to a conversation in which it participates. The probability to identify a relevance value remains constant, and is independent of everything else.
- The collector has no false-positive or false-negative errors.

C. THE UPDATING PROCESS

Screening a conversation might result in updating the distribution of one or more of the variables in the model, according to Bayes theory. We now describe the updating process after screening a conversation. We consider two possible situations prior to the screening: 1) Both nodes participating in the conversation are already identified, 2) At least one of the nodes is unidentified. First we consider the case where the two nodes are identified.

1. Updating When the Two Nodes are already Identified before the Screening

As mentioned before, if the nodes *i* and *j* are identified, then P_{ij} is independent of all the other variables (other than $S_{ij}^{(k)}$). Therefore, the result of screening a conversation between i and j would only result in updating the distribution of P_{ij}; it will not affect the PMF of the unidentified nodes

We assume that each P_{ij} has a probability distribution that belongs to the same family of distributions, with parameters determined by the values of D_i and D_j. The

family of distributions is chosen so as to allow a convenient way of updating the probabilities throughout the screening process. Specifically, we wish that the prior probability distribution of P_{ij} and its posterior distribution will belong to the same family of probability distributions, that is, we seek a *conjugate distribution.* The Beta distribution satisfies this property with respect to the Bernoulli likelihood functions [George at al., 1993] and its support is between 0 and 1, as desired. Therefore, we assume that for any (i, j) the probability distribution of P_{ij} given the values of D_i and D_j is a Beta distribution.

The PDF of P_{ij}, given the relevance values $D_i = d_i, D_j = d_j$, is: $f_{P_{ij}}(t \mid D_i = d_i,\ D_j = d_j) = \frac{t^{\alpha(d_i,d_j)-1}(1-t)^{\beta(d_i,d_j)-1}}{B(\alpha(d_i,d_j),\beta(d_i,d_j))}$ where $B(\alpha(d_i,d_j),\beta(d_i,d_j))$ is the Beta function and $\alpha(d_i,d_j),\beta(d_i,d_j)$ are the shape parameters whose values depend on the values of D_i and D_j.

The posterior of any Beta random variable $X \sim Beta(\alpha,\beta)$ with respect to a Bernoulli likelihood function is $Beta(\alpha+1,\beta)$ if a success is observed, and $Beta(\alpha,\beta+1)$ if a failure is observed. Therefore, given $D_i = d_{i,} D_j = d_j$ and the outcome $S_{ij}^{(1)}$ of the conversation , the posterior probabilities are:

$$f_{P_{ij}}(t \mid S_{ij}^{(1)} = 1, D_i = d_i, D_j = d_j) = \frac{t^{\alpha(d_i,d_j)}(1-t)^{\beta(d_i,d_j)-1}}{B(\alpha(d_i,d_j)+1,\beta(d_i,d_j))},$$

$$f_{P_{ij}}(t \mid S_{ij}^{(1)} = 0, D_i = d_i, D_j = d_j) = \frac{t^{\alpha(d_i,d_j)-1}(1-t)^{\beta(d_i,d_j)}}{B(\alpha(d_i,d_j),\beta(d_i,d_j)+1)} \qquad (2.2)$$

and generally:

$$f_{P_{ij}}(t \mid S_{ij}^{(1)} = x_1,\ldots,S_{ij}^{(k)} = x_k, D_i = d_i, D_j = d_j) = \frac{t^{\alpha(d_i,d_j)-1+s_{ij}}(1-t)^{\beta(d_i,d_j)-1+f_{ij}}}{B(\alpha(d_i,d_j)+s_{ij},\beta(d_i,d_j)+f_{ij})} \qquad (2.3)$$

where s_{ij} is the number of screened relevant conversations between i and j, and f_{ij} is the number of screened irrelevant conversations (recall that the screening process is error-free) .

Since the expected value of a Beta distribution with the parameters α, β is $\frac{\alpha}{\alpha+\beta}$ then based on equation (2.1), the posterior probability that the k-th conversation between i and j is relevant given that $S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_{k-1} j_{k-1}}^{(k-1)} = x_{k-1}$ is:

$$\Pr(S_{ij}^{(k)} = 1 \mid S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_{k-1} j_{k-1}}^{(k-1)} = x_{k-1}) = \frac{\alpha(d_i, d_j) + s_{ij}}{\alpha(d_i, d_j) + s_{ij} + \beta(d_i, d_j) + f_{ij}} \qquad (2.4)$$

where the number of relevant and irrelevant conversations, s_{ij} and f_{ij}, is determined according to the values of $S_{i_1 j_1}^{(1)} = x_{k-1}, ..., S_{i_{k-1} j_{k-1}}^{(k-1)} = x_{k-1}$.

This completes the updating process when both nodes are identified. If at least one of the nodes is unidentified, the updating process is more complicated, and requires the use of *graphical models*. We therefore start by providing some background about factors and graphical models.

2. Hammersley-Clifford Theorem

A graph of random variables is a graph in which nodes represent random variables, and edges represent dependencies between the random variables. Such graph holds the *Markov property* if every node is independent of all the other nodes in the graph given the values of its neighbors. In our model, the graph of the D_i have the same topology as the network graph, as if nodes i and j in the network graph are connected, their relevance values D_i and D_j are dependent. As mentioned before, this graph also holds the Markov property.

Hammersley-Clifford theorem (Hammersley and Clifford, 1971) states that if a graph of random variables holds the Markov property and the joint probability distribution of the random variables is strictly positive, then the joint distribution of all

the nodes in the graph can be represented by a normalized product of factors. In our model, the joint distribution of the relevance values can be represented by:

$$\Pr(\{D_i\}) = \Pr(D_1 = d_1, ..., D_N = d_N) = \frac{1}{z}\prod_{c \in C} \psi(D_c) \quad (2.6)$$

where C is the set of cliques in the graph of the relevance values, $\psi(D_c)$ is a factor of the random variables in clique c and z is a normalization factor. The product of the factors can be represented by a graphical model. We start by showing what factors are, and then explain what a graphical model is.

3. Factors

A *factor* represents dependencies between a set of random variables. It can be represented by a table, assigning each realization of the random variables a certain value. For example, a factor of binary random variables X, Y (denoted by $\psi(X, Y)$) might be represented by Table 1.

X	Y	Value
0	0	2
0	1	5
1	0	20
1	1	10

Table 2. Representation of a factor

In this example, if the value of X is 0, the value of Y is more likely to be 1 (as 5 is larger than 2); if the value of Y is 1, the value of X is more likely to be 1 (since 10 is larger than 5); etc. It is important to notice that a factor is not necessarily a representation of joint or conditional probability (it is often not normalized). In many cases, deriving the joint probabilities requires more than one factor (Koller & Friedman, 2010).

Basic operations with factors include reducing a factor given the value of one of its variables. For example, if the value of X is set to 1, the representation of the reduced factor $\psi(X = 1, Y)$ is shown in Table 3.

Y	Value
0	20
1	10

Table 3. Representation of a reduced factor

Another operation is marginalizing a factor over a certain variable, by eliminating the column of that variable and then summing up the identical rows. For example, the representation of the marginalized factor $\psi_X(Y)$ is shown in Table 4.

Y	Value
0	22
1	15

Table 4. Representation of a marginalized factor

We can also multiply factors, by creating a new factor containing all the variables of the factors. The values of this new factor are the multiplication of the appropriate values of the old factors. For example, the result of multiplying two factors is shown in Table 5.

X	Y	Value
0	0	2
0	1	5
1	0	20
1	1	10

Y	Z	Value
0	0	3
0	1	8
1	0	2
1	1	5

X	Y	Z	Value
0	0	0	6
0	0	1	16
0	1	0	10
0	1	1	25
1	0	0	60
1	0	1	160
1	1	0	20
1	1	1	50

Table 5. Up: two factors. Down: the product of the two factors

4. Graphical Models

A graphical model is a representation of dependencies amongst random variables. In our case, we use a graphical model called Markov Random Field (MRF). An MRF is an undirected graph whose nodes are factors, and there is an edge between two factors if and only if those factors share at least one random variable in common. For example, the factors $\psi(X_1, X_2), \psi(X_2, X_3)$ are connected. If there is only one factor in the product, it is proportional to the joint distribution of the nodes in the factor. However, this is not necessarily true if there is more than one factor.

We can use an MRF to determine the distribution of a subset of the relevance values (for example, the joint distribution of two relevance values (D_i, D_j)). We construct an MRF whose factors are $\{\psi(D_c)\}$ (equation (2.6)) which represent the different cliques in the network. Then, we use a method called *variable elimination* (Koller et Friedman, 2010). Variable elimination is an algorithm used to determine the joint distribution of a subset of the variables ("output variables"), given the assigned values of some of the other variables ("fixed variables"). In our case, if we want to determine the joint distribution of the variables $D_{i_1},...,D_{i_k}$, the input for the algorithm would be the identified relevance values, and the output would be a factor $\psi(D_{i_1},...,D_{i_k})$.

The first step in the algorithm is reducing all the factors which contain fixed variables (the way to reduce a factor is shown in the previous section). Then, at each iteration a random variable X ,which is neither a fixed nor an output variable is chosen. Then, X is eliminated by multiplying all the factors containing X and marginalizing the outcome over X. The process is repeated until only the output variables are left. The outcome of the algorithm does not depend on the order in which the variables are eliminated. However, since the run-time of the algorithm is determined by the order, several algorithms exist to choose an order which would minimize the run-time.

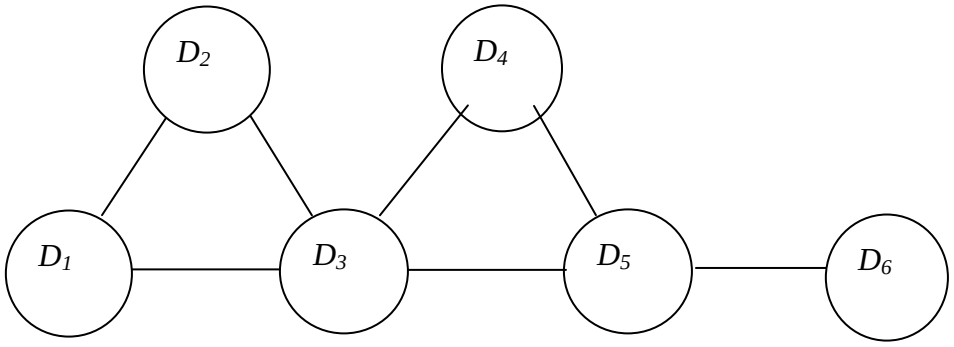


Figure 1. A graph to illustrate the variable elimination method

For example, suppose that given the graph in Figure 1, we wish to determine the joint distribution of (D_3, D_4). We start when we know the factors of the cliques in the graph, $\psi(D_1, D_2, D_3), \psi(D_3, D_4, D_5), \psi(D_5, D_6)$. Suppose that the value of $D_1 = d_1$ was identified. We therefore need to eliminate the random variables D_2, D_5, D_6 (the order of the elimination does not affect the outcome, but affects the run-time of the algorithm):

1) The initial MRF is: $\psi(D_1, D_2, D_3), \psi(D_3, D_4, D_5), \psi(D_5, D_6)$

2) We then reduce the factor $\psi(D_1, D_2, D_3)$ containing the identified relevance value D_1, and replace it with the reduced factor $\varphi(D_1, D_2, D_3) \triangleq \psi(D_2, D_3)$. The MRF after reducing D_1 is: $\varphi(D_2, D_3), \psi(D_3, D_4, D_5), \psi(D_5, D_6)$

3) To eliminate D_5, we multiply the factors containing D_5, such that $\psi(D_3, D_4, D_5) \cdot \psi(D_5, D_6) = \psi(D_3, D_4, D_5, D_6)$ and after marginalizing over D_5 we get: $\psi(D_3, D_4, D_6)$. The MRF is now $\varphi(D_2, D_3), \psi(D_3, D_4, D_6)$.

4) To eliminate D_6 we marginalize over $\psi(D_3, D_4, D_6)$ to get $\psi(D_3, D_4)$. The MRF is now $\varphi(D_2, D_3), \psi(D_3, D_4)$.

5) To eliminate D_2 we simply marginalize over $\varphi(D_2, D_3)$ to get $\varphi(D_2)$. The MRF is now $\varphi(D_3), \psi(D_3, D_4)$.

6) We then multiply $\varphi(D_3) \cdot \psi(D_3, D_4) = \varphi(D_3, D_4)$, and after normalizing it, $\varphi(D_3, D_4)$ represents the joint distribution of (D_3, D_4).

5. Updating According to the Relevance of the Conversation

According to equation (2.4), in order to determine the distribution of P_{ij}, we need to know the joint distribution of (D_i, D_j) and the number of relevant and irrelevant conversation between i and j. The collector therefore needs to update all joint distributions (D_i, D_j) for every edge (i, j) which contains at least one unidentified node. We start by ignoring the possibility that some of the nodes are identified, and we incorporate it into the updating process in section 6.

We begin with examining the simple case of updating the joint distribution of (D_i, D_j) after determining the relevance of a conversation between i and j, in the k th round.

The collector therefore knows that: $S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_{k-1} j_{k-1}}^{(k-1)} = x_{k-1}, S_{ij}^{(k)} = x_k$, where $x_1, ..., x_k \in \{0,1\}$. Using Bayes rule:

$$\Pr(D_i = d_i, D_j = d_j \mid S_{i_1 j_1}^{(1)} = x_1, ..., S_{ij}^{(k)} = x_k) =$$
$$\frac{\Pr(S_{ij}^{(k)} = x_k \mid D_i = d_i, D_j = d_j, S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_{k-1} j_{k-1}}^{(k-1)} = x_{k-1})}{\Pr(S_{ij}^{(k)} = x_k \mid S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_{k-1} j_{k-1}}^{(k-1)} = x_{k-1})} \Pr^{(k-1)}(D_i = d_i, D_j = d_j) \quad (2.7)$$

where $\Pr^{(k-1)}(D_i = d_i, D_j = d_j) \triangleq \Pr(D_i = d_i, D_j = d_j \mid S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_{k-1} j_{k-1}}^{(k-1)} = x_{k-1})$.

Based on equation (2.4), we know that:

$$\Pr(S_{ij}^{(k)} = 1 \mid D_i = d_i, D_j = d_j, S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_{k-1} j_{k-1}}^{(k-1)} = x_{k-1}) = \frac{\alpha(d_i, d_j) + s_{ij}^{(k)}}{\alpha(d_i, d_j) + s_{ij}^{(k)} + \beta(d_i, d_j) + f_{ij}^{(k)}}$$

$$\Pr(S_{ij}^{(k)} = 0 \mid D_i = d_i, D_j = d_j, S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_{k-1} j_{k-1}}^{(k-1)} = x_{k-1}) = \frac{\beta(d_i, d_j) + f_{ij}^{(k)}}{\alpha(d_i, d_j) + s_{ij}^{(k)} + \beta(d_i, d_j) + f_{ij}^{(k)}}$$

where $s_{ij}^{(k)}$ and $f_{ij}^{(k)}$ are the number of successes and failures amongst the screened conversation between i and j in the k th round. Therefore, expression (2.7) can be rewritten as:

$$\Pr^{(k)}(D_i = d_i, D_j = d_j \mid S_{ij}^{(k)} = 1) = \frac{\frac{\alpha(d_i,d_j) + s_{ij}^{(k)}}{\alpha(d_i,d_j) + s_{ij}^{(k)} + \beta(d_i,d_j) + f_{ij}^{(k)}}}{\sum_{d_i,d_j} \Pr^{(k-1)}(D_i = d_i, D_j = d_j) \frac{\alpha(d_i,d_j) + s_{ij}^{(k)}}{\alpha(d_i,d_j) + s_{ij}^{(k)} + \beta(d_i,d_j) + f_{ij}^{(k)}}} \Pr^{(k-1)}(D_i = d_i, D_j = d_j)$$

$$\Pr^{(k)}(D_i = d_i, D_j = d_j \mid S_{ij}^{(k)} = 0) = \frac{\frac{\beta(d_i,d_j) + f_{ij}}{\alpha(d_i,d_j) + s_{ij} + \beta(d_i,d_j) + f_{ij}}}{\sum_{d_i,d_j} \Pr^{(k-1)}(D_i = d_i, D_j = d_j) \frac{\beta(d_i,d_j) + f_{ij}}{\alpha(d_i,d_j) + s_{ij} + \beta(d_i,d_j) + f_{ij}}} \Pr^{(k-1)}(D_i = d_i, D_j = d_j)$$

(2.8)

We now address the more complicated problem of updating a joint distribution of two nodes, based on the relevance of a conversation between two other nodes. We update (D_i, D_j) after screening a conversation between nodes i_k and j_k, where i_k and j_k might be different than i and j. In order to do that, we will use a Graphical Model (shown in the previous section). According to the definition of conditional probability, and equation (2.6):

$$\begin{aligned}
&\Pr(D_1 = d_1, ..., D_N = d_N, S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_k j_k}^{(k)} = x_k) = \\
&\Pr(S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_k j_k}^{(k)} = x_k \mid D_1 = d_1, ..., D_N = d_N) \Pr(D_1 = d_1, ..., D_N = d_N) = \\
&\frac{1}{z'} \prod_{c \in C} \psi(D_c) \prod_{(i,j)} \Pr(S_{ij}^{ord(1)}, ..., S_{ij}^{ord(k_{ij})} \mid D_i, D_j)
\end{aligned} \tag{2.9}$$

where $\Pr(S_{ij}^{ord(1)}, ..., S_{ij}^{ord(k_{ij})} \mid D_i, D_j)$ is the joint distribution of all the conversations between i and j which were screened during the first k rounds. $ord(l)$ is the round in which the l th conversation was screened. The constant z' is used for normalization.

We can represent this joint distribution by adding to the MRF of the clique factors $\psi(D_c)$ (shown in the previous section) and the factors $\varphi_{ij}(S_{ij}^{ord(1)}, ..., S_{ij}^{ord(k_{ij})})$ which represent the joint probability $\Pr(S_{ij}^{ord(1)}, ..., S_{ij}^{ord(k_{ij})} \mid D_i, D_j)$, for all edge (i, j) which were sampled at least once during the first k rounds. As the number of conversations increases, the size of those factors grows exponentially. However, using the chain rule:

$$\Pr(S_{ij}^{ord(1)} = x_{ord(1)}, ..., S_{ij}^{ord(k_{ij})} = x_{ord(k_{ij})} \mid D_i = d_i, D_j = d_j) =$$
$$\Pr(S_{ij}^{ord(1)} = x_{ord(1)} \mid D_i = d_i, D_j = d_j) \Pr(S_{ij}^{ord(2)} = x_{ord(2)} \mid S_{ij}^{ord(1)} = x_{ord(1)}, D_i = d_i, D_j = d_j) \cdot ...$$

$$... \cdot \Pr(S_{ij}^{ord(k_{ij})} = x_{ord(k_{ij})} \mid S_{ij}^{ord(1)} = x_{ord(1)}, ..., S_{ij}^{ord(k_{ij}-1)} = x_{ord(k_{ij}-1)}, D_i = d_i, D_j = d_j) \qquad (2.10)$$

Each product in this multiplication can be easily calculated, using equation 2.7.

Using Variable Elimination, we can use the new MRF to calculate the joint distribution of every couple (D_i, D_j). The next section provides an illustration of this process.

6. Updating when Several Nodes in the Graph are Identified

Suppose that several nodes in the graph, without loss of generality $D_1, ..., D_m$, are identified, i.e., $D_1 = d_1, ..., D_m = d_m$ with probability of 1. Kohler et Friedman show a variation of the variable elimination algorithm (Kohler et Friedman, 2010) which can be used to determine the following expression:

$$\Pr(D_i = d_i, D_j = d_j \mid S_{i_1 j_1}^{(1)} = x_1, ..., S_{i_k j_k}^{(k)} = x_k, D_1 = d_1, ..., D_m = d_m) \qquad (2.11)$$

According to this method, a new MRF is constructed by reducing each factor in the original MRF which contains any of the random variables $D_1, ..., D_m$. Then, the variable elimination algorithm is used on the new MRF to determine the posterior joint probability of (D_i, D_j).

7. Example

We now show an example for the updating process. Since the graph in this example is very simple, one might use simple Bayesian updating. However, in more complicated graphs this might not be the case. Suppose we have the following graph in which each node has a relevance value of either 0 or 1, represented by the factor ψ (the factor happened to be normalized, although this is not necessary):

D_1	D_2	D_3	Value
0	0	0	0.2
0	0	1	0.1
0	1	0	0.1
0	1	1	0.1
1	0	0	0.1
1	0	1	0.1
1	1	0	0.1
1	1	1	0.2

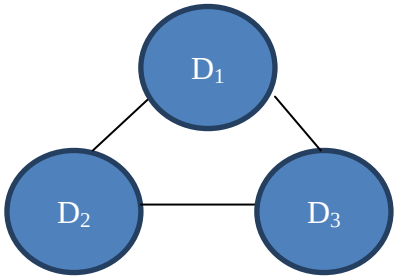


Figure 2. A graph and the factor ψ representing the dependencies in the graph. The graph and the factor are used to illustrate the updating process.

We wish to find the joint distribution of (D_2, D_3) after screening two conversations between nodes 1 and 2, given that $\alpha(0,0) = 0.5, \alpha(0,1) = \alpha(1,0) = 0.75,$ $\alpha(1,1) = 1$ and $\beta = 1$ (regardless of the relevance values). The prior distribution $P^{(0)}(D_2, D_3)$ is obtained by calculating the marginalized factor $\psi|_{D_1}$, as shown in Table 6.

D_2	D_3	Value
0	0	0.3
0	1	0.2
1	0	0.2
1	1	0.3

Table 6. The prior distribution of the factor

According to Table 6, we know that $\Pr(D_2 = 1) = \Pr(D_3 = 1) = 0.5$.

Suppose the first conversation is relevant. Then, we add to the graph the factor $\varphi_{12}^{(1)}(S_{12}^{(1)} = 1)$ (the factor is not normalized). The rows in which $S_{12}^{(1)} = 0$ were omitted, since their value is zero. The factor is represented by Table 7.

D_1	D_2	$S_{12}^{(1)}$	Value
0	0	1	$\frac{\alpha(0,0)}{\alpha(0,0)+\beta(0,0)}=\frac{0.5}{1.5}=\frac{1}{3}$
0	1	1	$\frac{\alpha(0,1)}{\alpha(0,1)+\beta(0,1)}=\frac{0.75}{1.75}=\frac{3}{7}$
1	0	1	$\frac{\alpha(1,0)}{\alpha(1,0)+\beta(1,0)}=\frac{0.75}{1.75}=\frac{3}{7}$
1	1	1	$\frac{\alpha(1,1)}{\alpha(1,1)+\beta(1,1)}=\frac{1}{2}$

Table 7. The factor after adding the fixed variable $S_{12}^{(1)}$

Then, we perform variable elimination, by eliminating first $S_{12}^{(1)}$ and then D_1. Clearly, the values of $\varphi_{12}^{(1)}(S_{12}^{(1)}=1)$ do not change when we eliminate $S_{12}^{(1)}$. For the second stage, we start by multiplying the factors $\varphi_{12}^{(1)}|_{S_{12}^{(1)}}$ and ψ.

D_1	D_2	D_3	Value
0	0	0	0.0667
0	0	1	0.033
0	1	0	0.043
0	1	1	0.043
1	0	0	0.043
1	0	1	0.043
1	1	0	0.05
1	1	1	0.01

Table 8. A multiplication of the two factors

Then, after marginalizing the factor by D_1 and normalizing the result, we get a new factor.

D_2	D_3	Value
0	0	0.26
0	1	0.18
1	0	0.22
1	1	0.34

Table 9. The updated factor given that the first conversation is relevant

Based on Table 9, $\Pr(D_2 = 1) = 0.56$ and $\Pr(D_3 = 1) = 0.52$. As expected, both probabilities are larger than the prior, and the probability of node 2 is larger than that of node 3. We can calculate this joint distribution using Bayes rule, as follows:

$$\Pr(D_2 = d_2, D_3 = d_3 \mid S_{12}^{(1)} = 1) = \frac{\Pr(S_{12}^{(1)} = 1 \mid D_2 = d_2, D_3 = d_3)}{\Pr(S_{12}^{(1)} = 1)} \Pr(D_2 = d_2, D_3 = d_3) =$$

$$= \frac{\sum_{d_1} \Pr(S_{12}^{(1)} = 1 \mid D_1 = d_1, D_2 = d_2) \Pr(D_1 = d_1 \mid D_2 = d_2, D_3 = d_3)}{\Pr(S_{12}^{(1)} = 1)} \Pr(D_2 = d_2, D_3 = d_3)$$

We would then get the exact same joint distribution.

Now, suppose the second conversation between 1 and 2 is irrelevant. We then construct an MRF with the factor ψ and a new factor, $\varphi_{12}^{(2)}(S_{12}^{(1)} = 1, S_{12}^{(2)} = 0)$.

D_1	D_2	$S_{12}^{(1)}$	$S_{12}^{(2)}$	Value
0	0	1	0	$\frac{1}{3}\frac{\beta(0,0)}{\alpha(0,0)+1+\beta(0,0)}=\frac{1}{7.5}$
0	1	1	0	$\frac{3}{7}\frac{\beta(0,1)}{\alpha(0,1)+1+\beta(0,1)}=\frac{3}{19.25}$
1	0	1	0	$\frac{3}{7}\frac{\beta(1,0)}{\alpha(1,0)+1+\beta(1,0)}=\frac{3}{19.25}$
1	1	1	0	$\frac{1}{2}\frac{\beta(1,1)}{\alpha(1,1)+1+\beta(1,1)}=\frac{1}{6}$

Table 10. The factor given that the second conversation was irrelevant

The variable elimination is very similar to the one performed in the previous iteration. We start by eliminating $S_{12}^{(1)}$ and $S_{12}^{(2)}$, which does not change the values of the factors. Then we multiply $\varphi_{12}^{(2)}|_{S_{12}^{(1)},S_{12}^{(2)}}$ and ψ.

D_1	D_2	D_3	Value
0	0	0	0.0266
0	0	1	0.0133
0	1	0	0.0156
0	1	1	0.0156
1	0	0	0.0156
1	0	1	0.0156
1	1	0	0.0166
1	1	1	0.0333

Table 11. The updated factor (not normalized)

After marginalizing over D_1, the normalized result is shown in Table 12.

D_2	D_3	Value
0	0	0.28
0	1	0.19
1	0	0.21
1	1	0.32

Table 12. The updated joint distribution given one relevant and one irrelevant conversations.

Based on this joint distribution, $\Pr(D_2 = 1) = 0.53$ and $\Pr(D_3 = 1) = 0.51$. As expected, this is a decrease regarding the previous values, where the highest decrease is for D_2. This result can also be achieved by calculating the following expression:

$$\Pr(D_2 = d_2, D_3 = d_3 \mid S_{12}^{(1)} = 1, S_{12}^{(2)} = 0) = \frac{\Pr(S_{12}^{(2)} = 0, S_{12}^{(1)} = 1 \mid D_2 = d_2, D_3 = d_3)}{\Pr(S_{12}^{(2)} = 0, S_{12}^{(1)} = 1)} \Pr(D_2 = d_2, D_3 = d_3) =$$

$$= \frac{\sum_{d_1} \Pr(S_{12}^{(2)} = 0 \mid S_{12}^{(1)} = 1, D_1 = d_1, D_2 = d_2) \Pr(S_{12}^{(1)} = 1 \mid D_1 = d_1, D_2 = d_2) \Pr(D_1 = d_1 \mid D_2 = d_2, D_3 = d_3)}{\Pr(S_{12}^{(2)} = 0 \mid S_{12}^{(1)} = 1) \Pr(S_{12}^{(1)} = 1)} \cdot$$

$$\cdot \Pr(D_2 = d_2, D_3 = d_3)$$

Now, we choose to screen a conversation between nodes 1 and 3, and find out that the conversation is relevant. The new MRF will include the factors ψ, $\varphi_{12}^{(2)}(S_{12}^{(1)} = 1, S_{12}^{(2)} = 0)$ and a new factor $\varphi_{13}^{(3)}(S_{13}^{(3)} = 1)$.

D_1	D_2	$S_{13}^{(3)}$	Value
0	0	1	$\frac{\alpha(0,0)}{\alpha(0,0)+\beta(0,0)} = \frac{0.5}{1.5} = \frac{1}{3}$
0	1	1	$\frac{\alpha(0,1)}{\alpha(0,1)+\beta(0,1)} = \frac{0.75}{1.75} = \frac{3}{7}$
1	0	1	$\frac{\alpha(1,0)}{\alpha(1,0)+\beta(1,0)} = \frac{0.75}{1.75} = \frac{3}{7}$
1	1	1	$\frac{\alpha(1,1)}{\alpha(1,1)+\beta(1,1)} = \frac{1}{2}$

Table 13. The factor givent a relevant conversation from a different edge

Performing variable elimination, we start by eliminating $S_{12}^{(1)}$, $S_{12}^{(2)}$ and $S_{13}^{(3)}$, without changing the values. Then, as before, we multiply the three fators, and marginalize the product by D_1. The normalized new factor which represents $\Pr^{(3)}(D_2, D_3)$ is shown in Table 14.

D_2	D_3	Value
0	0	0.24
0	1	0.21
1	0	0.19
1	1	0.36

Table 14. The updated distribution

The marginal probabilities are $\Pr(D_2 = 1) = 0.55$ and $\Pr(D_3 = 1) = 0.57$, as expected both values are higher, but this time the value for D_3 is higher than the value for D_2. The joint distribution can be calculated (and the results would be the same) using the expression:

$$\Pr(D_2 = d_2, D_3 = d_3 \mid S_{12}^{(1)} = 1, S_{12}^{(2)} = 0, S_{13}^{(3)} = 1) = \frac{\Pr(S_{13}^{(3)} = 1, S_{12}^{(2)} = 0, S_{12}^{(1)} = 1 \mid D_2 = d_2, D_3 = d_3)}{\Pr(S_{13}^{(3)} = 1, S_{12}^{(2)} = 0, S_{12}^{(1)} = 1)} \Pr(D_2 = d_2, D_3 = d_3) =$$

$$= \frac{\sum_{d_1} \Pr(S_{13}^{(3)} = 1 \mid D_1 = d_1, D_3 = d_3) \Pr(S_{12}^{(2)} = 0 \mid S_{12}^{(1)} = 1, D_1 = d_1, D_2 = d_2) \Pr(S_{12}^{(1)} = 1 \mid D_1 = d_1, D_2 = d_2) \Pr(D_1 = d_1 \mid D_2 = d_2, D_3 = d_3)}{\Pr(S_{12}^{(2)} = 0 \mid S_{12}^{(1)} = 1) \Pr(S_{12}^{(1)} = 1)} \cdot$$

$$\cdot \Pr(D_2 = d_2, D_3 = d_3)$$

Last, suppose the collector screens another conversation between nodes 1 and 3, the conversation turns out to be irrelevant, but based on the content of the conversation the collector determines that $D_1 = 1$ (i.e., node 1 was identified). As before, we start with the factors ψ, $\varphi_{12}^{(2)}(S_{12}^{(1)} = 1, S_{12}^{(2)} = 0)$ and $\varphi_{13}^{(4)}(S_{13}^{(3)} = 1, S_{13}^{(4)} = 0)$, represented by Table 15.

D_1	D_2	$S_{13}^{(3)}$	$S_{13}^{(4)}$	Value
0	0	1	0	$\frac{1}{3}\frac{\beta(0,0)}{\alpha(0,0)+1+\beta(0,0)} = \frac{1}{7.5}$
0	1	1	0	$\frac{3}{7}\frac{\beta(0,1)}{\alpha(0,1)+1+\beta(0,1)} = \frac{3}{19.25}$
1	0	1	0	$\frac{3}{7}\frac{\beta(1,0)}{\alpha(1,0)+1+\beta(1,0)} = \frac{3}{19.25}$
1	1	1	0	$\frac{1}{2}\frac{\beta(1,1)}{\alpha(1,1)+1+\beta(1,1)} = \frac{1}{6}$

Table 15. The factor given an irrelevant conversation from a different edge

Then, all factors are reduced given that $D_1 = 1$. The resulting factors are shown in Table 16.

D_1	D_2	$S_{13}^{(3)}$	$S_{13}^{(4)}$	Value
1	0	1	0	$\frac{3}{7}\frac{\beta(1,0)}{\alpha(1,0)+1+\beta(1,0)} = \frac{3}{19.25}$
1	1	1	0	$\frac{1}{2}\frac{\beta(1,1)}{\alpha(1,1)+1+\beta(1,1)} = \frac{1}{6}$

D_1	D_2	$S_{12}^{(1)}$	$S_{12}^{(2)}$	Value
1	0	1	0	$\frac{3}{7}\frac{\beta(1,0)}{\alpha(1,0)+1+\beta(1,0)} = \frac{3}{19.25}$
1	1	1	0	$\frac{1}{2}\frac{\beta(1,1)}{\alpha(1,1)+1+\beta(1,1)} = \frac{1}{6}$

D_1	D_2	D_3	Value
1	0	0	0.1
1	0	1	0.1
1	1	0	0.1
1	1	1	0.2

Table 16. The three remaining factors

Now, after eliminating $S_{12}^{(1)}, S_{12}^{(2)}, S_{13}^{(3)}, S_{13}^{(4)}$, multiplying the remaining factors and eliminating D_1, the normalized result would be:

D_2	D_3	Value
0	0	0.18
0	1	0.2
1	0	0.2
1	1	0.42

Table 17. The updated factor

As expected, now both nodes have the same probability to have a relevance value of 1, $\Pr(D_2 = 1) = \Pr(D_3 = 1) = 0.62$, higher than the prior probability because of the identification of node 1.

III. ALGORITHMS AND HEURISTICS

In this chapter we propose several algorithms and heuristics to address the information selection problem described in Chapter I and modeled in Chapter II. Each algorithm employs a different strategy for choosing conversations for screening, in order to maximize the expected number of relevant conversations screened. The performance of these algorithms and heuristics is described in the next chapter.

A. THE OPTIMAL STRATEGY

Theoretically, the optimal strategy which maximizes the expected number of relevant conversations can be obtained using Partially Observable Markov Decision Process (Cassandra et al., 1994; Boutilier, 2002), as will be explained next. The following analysis is similar to that shown by Frazier et al. (Frazier et al., 2009).

We first need to distinguish between the state of the world and the belief regarding this state. The *state of the world* is a vector of the values d_i, p_{ij}, where d_i is the true relevance value of node i and p_{ij} is the probability that a conversation between i and j is relevant. Formally, the state of the world is denoted by $w = (\overline{d}, \overline{p})$, where $\overline{d} = (d_1, ..., d_N), \overline{p} = (p_{i_1 j_M}, ..., p_{i_M j_M})$. According to our assumptions (stated in chapter II) the state of the world does not change. As mentioned in chapter II, the collector does not know what the state of the world is. The collector only has a probability distribution over possible states of the world, and he updates this distribution throughout the screening process as new knowledge is gained through the screening of the conversations.

We can therefore define the *state of the collector* as the information gained by the collector throughout the screening process. The state of the collector is represented by a vector of the number of relevant and irrelevant conversations screened from each edge, and the identified relevance values. As shown in the updating process section (in Chapter II), with the prior joint distribution, this information is sufficient for updating the joint probability distribution of the relevance values ($\Pr(D_1 = x_1, ..., D_N = x_N)$). Therefore, the

information is also sufficient to describe the joint density function of the different P_{ij}. Formally, the state of the collector is the tuple $r^{(k)} = (\overline{s^{(k)}}, \overline{f^{(k)}}, \overline{d^{(k)}})$, where:

$$\overline{s^{(k)}} = (s_{i_1 j_1}{}^{(k)}, ..., s_{i_M j_M}{}^{(k)}), \overline{f^{(k)}} = (f_{i_1 j_1}{}^{(k)}, ..., f_{i_M j_M}{}^{(k)}), \overline{d^{(k)}} = (d_1{}^{(k)}, ..., d_N{}^{(k)}).$$

As defined in Chapter II, $s_{ij}{}^{(k)}$, $f_{ij}{}^{(k)}$ are the numbers of relevant and irrelevant conversations screened from edge (i, j) during the first k rounds. $d_i{}^{(k)}$ is a categorical parameter, whose value is $d_i{}^{(k)} = d_i$ if the relevance value has been identified during the first k rounds, and $d_i{}^{(k)} = "null"$ otherwise. We now define actions, strategies, rewards and transition probabilities. An action is simply screening a conversation between i and j. Therefore, each action can be represented by the tuple (i, j), and the set of possible actions at round k is all the edges which still have unscreened conversations. Formally, the set of possible actions is $A(r^{(k)}) = \{(i, j) \mid n_{ij}{}^{(k)} > 0\}$ where $n_{ij}{}^{(k)}$ is the number of unscreened conversations between i and j at the k th round. We can therefore define a strategy π as a rule for choosing an action given a state *r*. The strategy takes into account the state of the collector, not the state of the world . The collector receives a reward of 1 if a relevant conversation is screened and 0 otherwise.

Given a state $r^{(k)}$, if the chosen action in the *(k+1)*th round is to screen a conversation from edge (i, j), then state $r^{(k)}$ might change in the following ways:

1) The number of relevant and irrelevant conversations from edge (i, j) in state $r^{(k+1)}$ can be either $(s_{ij} + 1, f_{ij})$ or $(s_{ij}, f_{ij} + 1)$, with probability p_{ij} and $1 - p_{ij}$, respectively.

2) If the relevance value of one of the nodes is still unidentified, w.l.o.g. i (i.e., $d_i{}'^{(k)} = "null"$), then the relevance value d_i might be identified by screening the conversation, and as a result $d_i{}'^{(k+1)} = d_i$, with probability c, defined in Chapter II.

3) If both relevance values are unidentified, then only one of them is identified, with probability $c(1-c)$. The probability that both nodes are identified is c^2 and the probability that none of them are identified is $(1-c)^2$.

The transition probabilities clearly depend on the state of the world w. Since the probabilities that a conversation is relevant and the probability that the relevance value is identified are independent, the transition probabilities $\Pr(r^{(k+1)} \mid r^{(k)}, w, (i,j))$ is the product of p_{ij} or $(1-p_{ij})$ with either $1, c, (1-c), c^2, c(1-c)$ or $(1-c^2)$. We can therefore calculate the transition probability, i.e the probability that state $r^{(k)}$ would change into state $r^{(k+1)}$ following screening a conversation between i and j:

$$\Pr(r^{(k+1)} \mid r^{(k)}, (i,j)) = \int_w \Pr(r^{(k+1)} \mid r^{(k)}, w, (i,j)) \Pr(w \mid r^{(k)}) dw \qquad (3.1)$$

The expression $\Pr(w \mid r^{(k)})$ is obtained using equation (2.11)

Since the collector has only an estimate of the state of the world, we cannot use the conventional Bellman equation (Bellman, 1957) to determine the optimal policy. However, this problem can be formulated as a POMDP – partially observable Markov decision process (Cassandra et al., 1994). A POMDP problem includes the state of the world w, and a belief state $b(w)$, the estimated probability that the state of the world is w. The *value function* of a belief state b represents the expected reward if the optimal strategy (according to belief state b) is employed. It is determined using the recursive formula (Cassandra et al., 1994):

$$V^{(n)}(b) = \max_a \{ \sum_w b(w) R(a,w) + \sum_{b'} \tau(b,a,b') V^{(n+1)}(b') \} \qquad (3.2)$$

where a is an action, $R(a,w)$ is the expected reward given the state of the world w if action a was chosen, and $\tau(b,a,b') = \sum_w \Pr(b' \mid b,a,w) b(w)$ is the transition probability from belief state b to b' given that action a was chosen.

We can translate the notations in equation 3.2 into our model's terminology in the following manner:

- In both cases, the state of the world is denoted by w, although in our case w is continuous.
- As mentioned before, an action a is screening a conversation between i and j.
- The belief state $b(w)$ is represented by the probability $\Pr(w \mid r^{(k)})$.
- The reward $R(a,w)$ is the expected value of $S_{ij}^{(k)}$. Since that expected value is equal to the probability that the conversation is relevant, that expected value is simply p_{ij}.
- The transition probability $\tau(b,a,b')$ can be substituted with the transition probability $\Pr(r^{(k+1)} \mid r^{(k)},(i,j))$ in expression (3.1).

In the last round, the collector will simply choose the conversation with the highest probability to be relevant. Formally, the future value of the last round, given the state of collector $r^{(T-1)}$ is simply $V^{(T-1)}(r^{(T-1)}) = \max_{(i,j)}\{\mathrm{E}[S_{ij}^{(T)}]\}$. We can therefore use equation (3.2) recursively to calculate the future value in a given iteration and a state of the collector.

To determine which conversation should be chosen in each iteration, we can use equation (3.2), where the expression $\sum_{w} b(w)R(a,w)$ is substituted by $\int_{w} \Pr(w \mid r^{(k)})\, p_{ij}\, dp_{ij}$ which equals (according to equation (2.8)) $\mathrm{E}[S_{ij}^{(k)}]$. We therefore end up with the following equation to determine the best edge from which a conversation should be screened:

$$(i^*, j^*) = \arg\max_{(i,j)}\{\mathrm{E}[S_{ij}^{(k+1)}] + \sum_{r^{(k+1)}} \Pr(r^{(k+1)} \mid r^{(k)},(i,j))V^{(k+1)}(r^{(k+1)})\} \qquad (3.3)$$

Therefore, the strategy which chooses a conversation according to equation (3.3) is the optimal strategy.

Although we can theoretically determine the optimal strategy, for more than a few iterations this method is impractical, since the required number of calculations grows exponentially with the number of iterations. We therefore examine different approximate algorithms to provide us with a strategy as close to optimal as possible.

B. ALGORITHMS AND HEURISTICS

In this section we describe the different algorithms that we examine in Chapter V. We start by describing two basic approaches that are mentioned in the literature (Daw at al, 2006; Tokic, 2010) as common algorithms for handling the exploitation-exploration problem: Softmax and ε-greedy.

1. Basic Algorithms

The following two basic algorithms provide a baseline for comparison with more advanced algorithms developed in this chapter. Both algorithms run for a fixed number of iterations, and in each iteration choose one conversation to be screened. We define an *alternative* as an edge from which a conversation might be chosen, in other words, each edge with unscreened conversations is an alternative. The ε-greedy algorithm determines in each iteration whether to choose an alternative according to an exploitation criterion or an exploration criterion (explained below). The Softmax algorithm does not make this clear distinction—in each iteration it assigns weights to the different alternatives and chooses randomly according to the weights, thus combining exploration and exploitation.

a. Softmax

At each iteration the Softmax algorithms (Thrun, 1992) chooses one of several alternatives. A chosen alternative a is expected to produce a reward v_a. In our context, the alternatives are edges with unscreened conversations, and the expected reward is $\mathrm{E}[P_{ij}]$. In each iteration an alternative with a higher expected reward (i.e., higher $\mathrm{E}[P_{ij}]$) is more likely to be chosen. However, there is still a probability that alternatives with lower expected rewards are chosen.

The algorithm assigns each alternative a specific weight between 0 and 1 , which is designated as the probability that the alternative is chosen. The weights are assigned based on the expected rewards according to the Bolzman distribution formula: $w_a = \frac{e^{\frac{v_a}{T}}}{\sum_a e^{\frac{v_a}{T}}}$. T is a positive parameter called *temperature* (Daw at al., 2006). For small values of T , the weight of variables with high expected value is very large and they will almost always be chosen. For large values of T , all variables have about the same weight.

b. ε *-Greedy*

In the ε -greedy algorithm (Barton at al., 1998), each round an exploration approach is chosen with a probability of epsilon, and an exploitation approach otherwise. The purpose of exploration is to get more information on the different possible alternatives. Specifically, in the ε -greedy algorithm, exploration means choosing an alternative at random out of all the possible alternatives (i.e., out of all the edges with at least one unscreened conversation). Exploitation, however, means choosing an alternative which would maximize the expected reward. Specifically, in the following algorithms exploitation means choosing at random from some well-defined subset of top alternatives, i.e., alternatives with high values. The value of an alternative in our case (an edge) is $E[S_{ij}]$.

The value of ε might be constant, or a function of the number of iterations left for the algorithm (Tokic, 2010). For example, ε can be chosen to be $\varepsilon(t) = (1 - \frac{t}{T})^{\rho}$ where T is the total number of rounds (given at the beginning of the process), t is the current iteration, and ρ is a scaling parameter. The larger ρ is the faster the function decreases, and ε is 1 at the beginning of the process and 0 at the end. Therefore, exploration is more likely during the first iterations, while exploitation is more likely during the last rounds.

c. Pure Exploitation

One intuitive approach is to examine a greedy algorithm that always chooses a conversation from the edge with the highest expected probability to be a relevant conversation, i.e., the highest $E[P_{ij}]$. In other words, this algorithm ignores exploration, and always chooses a conversation according to the exploitation criterion. We will use this naïve approach as a baseline for comparison with the other algorithms.

d. Exploration-First Heuristic

Before addressing more complicated algorithms, we describe some naïve heuristics for solving the problem that are intuitively appealing and therefore might be employed by a collector. One such heuristic is to start with an exploration period, i.e., the purpose for choosing the first conversations is to gain information on the different alternatives, and then continue with an exploitation period, in which the goal in each iteration is to maximize the expected probability that the chosen conversation is relevant.

During the exploration period, the collector can use different exploration methods, such as the knowledge gradient policy and the wide exploration policy, both described later. During the exploitation period, the collector either always chooses the best alternative, i.e., the edge with the highest $E[P_{ij}]$, or chooses according to the Softmax algorithm.

e. A Naïve Exploration Method – Wide Exploration

An intuitive way for exploring the graph is to sample as many different edges as possible, rather than further evaluate the already sampled edges. Given an integer B, the collector would choose to explore the edge with the highest expected value, as long as it has been chosen less than B times so far.

2. Advanced Algorithms

a. ε-Greedy VDBE-Bolzman

The Value-Difference-Based-Exploration (VDBE) algorithm presented by Tokic (Tokic, 2010) is a modification of the ε-greedy algorithm, with a different

decision rule for determining whether to explore or to exploit. The algorithm assumes that an exploration criterion is more likely to be chosen when there is a low certainty regarding the expected values of the alternatives, and an exploitation criterion is applied otherwise. As mentioned before, an alternative is an edge from which a conversation might be screened, and the expected value of an alternative is $\mathrm{E}[P_{ij}]$. There is a low certainty regarding the expected values, if the expected values rapidly change after screening a conversation. Therefore, ε is determined according to the amount of change in the expected value of the chosen alternative.

In order to accommodate that, ε is being updated according to the formula:

$$\varepsilon^{(k+1)} = \delta \frac{1-e^{\frac{-\left|v_i^{(k+1)}-v_i^{(k)}\right|}{\sigma}}}{1+e^{\frac{-\left|v_i^{(k+1)}-v_i^{(k)}\right|}{\sigma}}} + (1-\delta)\varepsilon^{(k)} \qquad (3.4)$$

where i is the chosen alternative in the k th round, and $\left|v_i^{(k+1)}-v_i^{(k)}\right|$ is the change in the expected reward of alternative i (defined earlier as the expectation of the respective P_{ij}). The parameter σ is a positive constant called *inverse sensitivity*. The smaller σ is, the larger the impact a change in the expectation has on the value of epsilon. δ is another scaling parameter, which the way to determine it is explained later. The value of $\varepsilon^{(0)}$ is set to be 1. In our model terminology, $v_i^{(k+1)} - v_i^{(k)} = E[P_{ij}^{(k+1)}] - E[P_{ij}^{(k)}]$.

b. The Knowledge-Gradient Policy

Frazier et al. (Frazier et al., 2009) propose a solution for the following ranking and selection (R&S) problem. A decision maker is presented with several actions, each of which returns a random reward. The rewards are correlated and the decision maker's problem is to select the best action, i.e., the action with the highest average reward. Specifically, after alternative i is chosen, it produces rewards according to a Gaussian distribution whose mean and standard error are θ_i and σ_i respectively. The standard errors are known to the decision maker, but the means are unknown. However, it

is known that the different θ_i are drawn according to one multivariate normal Gaussian distribution whose parameters are unknown. The different θ_i are therefore correlated, and information about one of them provides information on the distribution of the others. The goal of the decision maker is to assess which action has the maximal θ_i. In order to do that, several rounds of exploration are allowed in which the different actions are sampled and evaluated.

In our terminology, Frazier et al. focus only on the exploration phase. They propose an algorithm that samples the different alternatives, and eventually determines what alternative has the highest expected value. They propose the Knowledge-Gradient (KG) policy to solve the problem, and show that it is the best myopic strategy possible (although non-myopic strategies might prove better).

The symbol $b^{(k)}$ denotes the belief state of the decision maker in the k th iteration (defined in Section A), i.e., its assessment of the different θ_i. Based on $b^{(k)}$, the expected value of the best alternative is denoted by $\theta_{\max}{}^{(k)} \mid b^{(k)}$. Following the sampling of an alternative a and observing the reward r from choosing it, the belief state of the decision maker changes into $b^{(k+1)} \mid r, a$, resulting with $\theta_{\max}{}^{(k+1)} \mid b^{(k)}, a, r$. Since the decision maker has an assessment regarding the distribution of r, he can estimate $\mathrm{E}[\theta_{\max}{}^{(k+1)} \mid a]$ for each alternative. According to the KG-policy, he chooses the alternative according to: $\arg\max_a \{\mathrm{E}[\theta_{\max}{}^{(k+1)} \mid a] - \theta_{\max}{}^{(k)}\}$. In other words, he chooses the alternative that is expected to change the most the maximal expected reward. We will now show the adaptation of this algorithm to our model.

There are two main differences between the model provided by Frazier et al. and our model. First, the parameters in Frazier's model have a joint multivariate normal distribution, while our parameters are also correlated, but in a way determined by the network structure of the conversation records. Second, Frazier et al. focus only on the exploration stage. They ignore the rewards gathered during the exploration portion. In our model, there is no clear distinction between an "exploration phase" and "exploitation

phase." Instead, the collector simply collects the maximum number of relevant conversations given his time constraint. Thus, any separation between "exploration" and "exploitation" is purely algorithmic and does not originate in the problem statement. Despite those differences, we can still use the KG policy as an exploration method.

Given a state of the collector $r^{(k)}$ (defined in section A), the collector estimates the value of $\mathrm{E}[P_{ij}]$ for every (i, j). Suppose that from the *k*th iteration onward, the collector chooses conversations based solely on the different values of $\mathrm{E}[P_{ij}]$ (without updating them), regardless of the outcomes of the following rounds. A greedy strategy would be to screen conversations from the edge with the highest $\mathrm{E}[P_{ij}]$ until it has no more conversations, then screen conversations from the edge with the second highest $\mathrm{E}[P_{ij}]$, and so on. The *future value* of a state $r^{(k)}$, denoted by $Q^{(k)}(r^{(k)})$, is the expected number of relevant conversations given the greedy strategy.

The future value is therefore the number of relevant conversations the collector expects to screen from the k th iteration onward, given $r^{(k)}$. Now, suppose that on the (*k*+1)th iteration the collector screens a conversation between i and j, determines $r^{(k+1)} \mid r^{(k)}, S_{ij}^{(k+1)}$ and only then employs the aforementioned greedy strategy. The expected number of relevant conversations screened from the k th iteration onward would then be: $S_{ij}^{(k+1)} + Q^{(k+1)}(r^{(k+1)})$. Then, the expression $\Delta_{ij}^{(k)} = S_{ij}^{(k+1)} + Q^{(k+1)}(r^{(k+1)}) - Q^{(k)}(r^{(k)})$ describes the change in the total expected reward from the k th round onward following the screening of a conversation between i and j. Taking an expectation, the expression becomes:

$$\mathrm{E}[\Delta_{ij}^{(k)}] = \mathrm{E}[S_{ij}^{(k+1)}] + \sum_{r^{(k+1)}} [Q^{(k+1)}(r^{(k+1)}) - Q^{(k)}(r^{(k)})] \cdot \Pr(r^{(k+1)} \mid r^{(k)}, (i, j)) \qquad (3.6)$$

The expression $\Pr(r^{(k+1)} \mid r^{(k)}, (i, j))$ is calculated according to equation (3.1).

According to the KG policy, the collector would choose at each iteration the edge with the highest expected change: $(i^*, j^*) = \arg\max_{(i,j)} \{E[\Delta_{ij}^{(k)}]\}$. For the last two rounds, the KG is the optimal policy.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ANALYSIS

A. SIMULATION DESCRIPTION

1. Overview

In order to test and compare the performance of the different algorithms and heuristics described in Chapter III, we have constructed a simulation of the screening process. We now show an overview of the way the simulation represents the state of the world and the state of the collector.

The network representing the state of the world consists of:

- A graph representing the communication network.
- The number of conversations n_{ij} between any two nodes (i,j) in the graph.
- The relevance value d_i assigned to each node i, where $d_i = 0$ if the node is irrelevant, and the probability p_{ij} that a conversation between nodes i and j is relevant for each edge (i, j).

The collector's knowledge and beliefs regarding the state of the world are:

- The collector knows the network's topology and the number n_{ij} of conversations between each pair of nodes (i,j).
- The collector does not know the true values d_i and p_{ij}, and therefore estimates them using the random variables D_i, P_{ij}. He has a prior joint probability distribution representing his belief regarding the different D_i. Based on that prior distribution and the conditional probabilities $\Pr(P_{ij} \mid D_i, D_j)$ known to him, he has a prior distribution of the P_{ij}.

- The collector updates the probability distributions of D_i and P_{ij} based on the observed relevance of the screened conversations. He keeps track of the number of relevant and irrelevant conversations screened from each edge, and the identified relevance values that may be revealed during the screening process.

The main stages of the simulation are:

- Stage 1: Creating a graph representing of the network;
- Stage 2: Determining the prior joint distributions of D_i and P_{ij};
- Stage 3: Setting the fixed values of the parameters d_i and p_{ij};
- Stage 4: Implementing a certain screening algorithm
    - Selecting an edge for screening,
    - Determining the outcome of the screening (based on the p_{ij} ,values determined in Stage 3),
    - Updating the state of the collector knowledge accordingly.

2. Stage 1 – Constructing the Network Graph

a. Main Assumptions

- We define a set of nodes in the graph that consists of nodes representing relevant persons in the network (those with d_i>0), and nodes representing irrelevant persons (d_i = 0).
- The edges between two nodes, each representing a relevant person, are given as input, that is…. The other edges (between nodes where at least one is irrelevant (d_i = 0)), are determined randomly. The number of conversations associated with a certain edge is determined by a Poisson distribution. The mean of the Poisson distribution is given as a parameter, and this value is the same for all edges.

b. Stage Description

We construct a graph in which each node represents a person, and there is an edge between two nodes if and only if there has been at least one conversation between the two respective persons.

The nodes in the network are divided into nodes representing relevant persons and nodes representing irrelevant persons. The total number of nodes is N. The set of edges between relevant persons is given. Edges connecting nodes representing irrelevant persons with either relevant or irrelevant persons are added randomly, as in an Erdos-Renyi graph (Erdos at Renyi, 1959): For each irrelevant node i, and another node j (either relevant or irrelevant) there is a predetermined probability that nodes i and j are connected.

After the edges are set, the number of conversations between two connected nodes i and j, n_{ij}, is determined by a number drawn from a Poisson distribution with a given mean, plus 1. The extra conversation added to the drawn number guarantees that there is at least one conversation for each edge.

c. Example

Given a graph representing connections between six relevant persons –

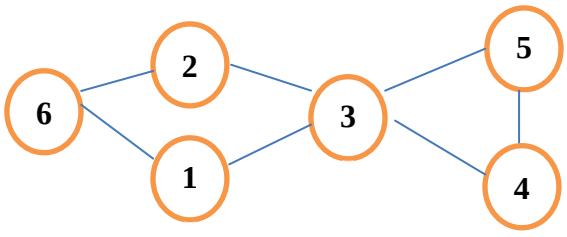


Figure 3. An example of a network

After adding nodes representing four irrelevant persons, adding randomly generated edges, and determining n_{ij} based on a Poisson distribution with mean 10, the resulted graph is shown in Figure 4.

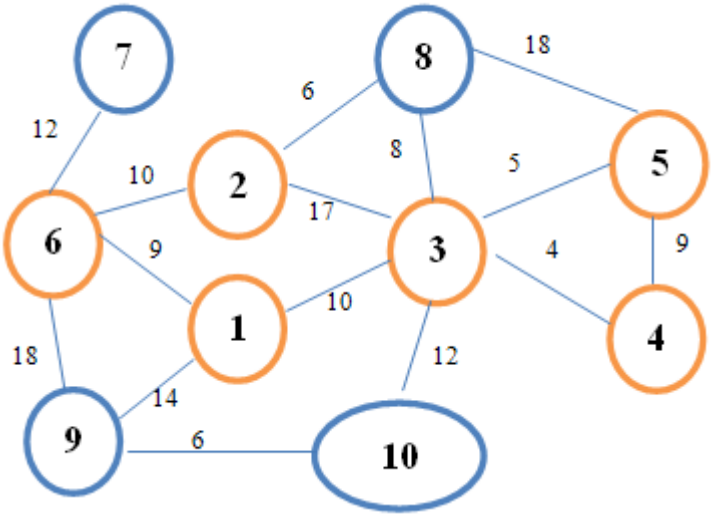


Figure 4. An example of a network with dummy nodes

3. Stage 2 – Determining the Distributions of the Random Variables

a. Main Assumptions

- Neighboring nodes are more likely to have similar relevance values.
- The probability distributions of the relevance values (i.e., $\Pr(D_i = d_i)$) are strictly positive, i.e., $\Pr(D_i = d_i) > 0$ for every i, d_i. This is a condition for Hammersely-Clifford theorem.
- The relevance value of a node (D_i) is independent of the relevance values of other nodes in the graph, given the relevance values of its neighbors. Therefore, the joint distribution of the relevance values can be represented by a product of joint distributions of the cliques in the network.
- Given relevance values d_i, d_j of two adjacent nodes (here d_i, d_j represent any values of the random variables D_i, D_j,- not necessarily the true relevance values drawn in Stage 3 below) the probability distribution of P_{ij} is a Beta distribution with the parameters $\alpha(d_i, d_j), \beta(d_i, d_j)$.

b. Stage Description

In order to apply the updating process described in Chapter II, we need to:

- Identify the cliques in the graph constructed in stage 1.
- Determine the clique factors $\psi(D_c)$.
- Determine the functions $\alpha(d_i, d_j), \beta(d_i, d_j)$, which determine the conditional probabilities $\Pr(P_{ij} \mid D_i = d_i, D_j = d_j)$.

(1) Identifying the Cliques in the Graph. As mentioned in Chapter II, in order to determine the joint distribution of the D_i and the P_{ij} we rely on Clifford-Hemersley theorem. To use that theorem we first need to determine the cliques in the graph constructed in Stage 1. The cliques of the graph are found according to Bron-Kerbosch algorithm (Bron et Kerbosch, 1973). For example, the cliques in the graph depicted in Figure 2 are: {1,6,9}, {6,7}, {2,6}, {2,3,8}, {1,3}, {3,10}, {3,4,5},{3,5,8},{9,10}.

(2) Clique Factors. As mentioned in Chapter II, we assume homophility in the network, that is, people with a high relevance value are more likely to be engaged in a conversation with other people of this type. Likewise, we assume that irrelevant people are more likely to communicate with other irrelevant people. Therefore, the relevance values of neighboring nodes are likely to be similar.

Let us assume we have a clique with m nodes and the relevance value of each node in the clique is one of l possible values. Then, the relevance values of the nodes have l^m possible realizations, where each realization is an m-dimensional vector. Given a realization $y = (y_1, ..., y_m)$, we can define a *weight* to represent how different are the values of the realization from each other: $w(y) = \sum_{i=1}^{m} (y_i - \bar{y})^2 + 1$, where $\bar{y} = \frac{1}{m}(\sum_{i=1}^{m} y_i)$ is the average of the different relevance values in the realization. One is added to avoid dividing by zero later on. If the values of a realization $y = (y_1, ..., y_m)$ are

close to each other, the weight of y would be low. If there is a significant variability among some of these values, the weight of y will be high. According to our homophily assumption, the higher the weight of a realization, the less likely it is to happen. We assign each realization a value $v(y)$ representing how likely the realization is. The higher the value, the higher the probability that the relevance values of the node are indeed $y_1,...,y_m$. The value assigned for each realization is $v(y) = (1/w(y))^p$ where p is a positive scaling parameter. When p equals zero, all realizations have the same probability, and when p is very large the probability of high- weight realizations is close to zero.

(3) Determining the Beta distributions. The probability distribution of P_{ij} depends on the relevance values of nodes i and j. As described in Chapter II and mentioned above,, for given relevance values d_i, d_j the distribution of P_{ij} is determined according to a Beta distribution with parameters $\alpha(d_i, d_j)$ and $\beta(d_i, d_j)$. We assume that the higher the relevance values of the nodes i and j, the higher the probability that a conversation between i and j is relevant (P_{ij}). The mean value of P_{ij} is $\frac{\alpha(d_i, d_j)}{\alpha(d_i, d_j) + \beta(d_i, d_j)}$ (the mean value of a Beta distribution). Therefore, if $\beta(d_i, d_j)$ remains constant, then the higher $\alpha(d_i, d_j)$ the higher the mean value of P_{ij}. We therefore assume here that while $\alpha(d_i, d_j)$ is an explicit function of d_i and d_j, $\beta(d_i, d_j)$ is constant. That is, $\alpha(d_i, d_j) = \frac{(d_i + 0.5)^q + (d_j + 0.5)^q}{2(\max_i\{d_i\} + 0.5)^q}$, and $\beta(d_i, d_j) = \beta$, where q is a scaling parameter. The value 0.5 is added to the relevance values to make sure that $\alpha(d_i, d_j) \neq 0$. If q is very high, then when d_i, d_j are low, $\Pr(P_{ij} = t)$ approaches zero for $t > 0$. If q is close to zero, then P_{ij} is independent of d_i, d_j. The function $\alpha(d_i, d_j)$ was chosen to be a monotone increasing function such that $\alpha(\max\{d_i\}, \max\{d_j\}) = 1$. The mean value of P_{ij} is therefore $\mathrm{E}[P_{ij} \mid D_i = D_j = \max\{d_i\}] = 0.5$.

(4) Initializing the Prior Joint Probabilities of (D_i, D_j). As mentioned in Chapter III, in order to determine the probability distributions of $P_{ij}, S_{ij}^{(k)}$ we need to know:

- The joint distribution of (D_i, D_j);
- The number of relevant and irrelevant conversations on edge (i, j)
- The Graphical model of the clique factors $\{\psi(D_c)\}$

Therefore, at the end of this stage, a Markov Random Field (MRF), composed of the Clique factors, is constructed. Based on this MRF, the simulation uses variable elimination to determine the joint distributions of (D_i, D_j). Those joint distributions are updated during the screening process. Finally, a table is constructed to keep track of the relevant and irrelevant conversations screened at each edge.

c. Example

The Graphical Model for the graph shown in Figure 4 includes the factors: $\psi(1,6,9), \psi(6,7), \psi(2,6), \psi(2,3,8), \psi(1,3), \psi(3,10), \psi(3,4,5), \psi(3,5,8), \psi(9,10)$, corresponding to the nine cliques identified in the graph.

Given the clique (2,3), and assuming that there are t three possible relevance values – 0,1 and 2, and both scaling parameters p and q equal 1, Table 18 shows the values of the factor $\psi(D_2, D_3)$, and the values of the α parameter for the Beta distribution.

D_2	D_3	Weight ($w(y)$)	Value ($v(y)$)	value of α
0	**0**	1	0.157895	0.2
0	**1**	1.5	0.105263	0.4
0	**2**	3	0.052632	0.6
1	**0**	1.5	0.105263	0.4
1	**1**	1	0.157895	0.6
1	**2**	1.5	0.105263	0.8
2	**0**	3	0.052632	0.6
2	**1**	1.5	0.105263	0.8
2	**2**	1	0.157895	1

Table 18. An example of the alpha function

4. Stage 3 – Drawing the Fixed Values

a. Key Assumptions

The fixed values p_{ij}, d_{ij} are randomly drawn from the joint distributions of the random variables P_{ij}, D_i known to the collector.

b. Stage Description

The fixed values d_i, for each node representing a relevant person, are assigned sequentially, based on the MRF determined in the previous section. . For each node i, which represents a relevant person, we use variable elimination to derive from the MRF the probability distribution of D_i. The value d_i is drawn from that distribution. Then, the MRF is marginalized according to the result, as shown in Chapter II. We keep a copy of the original MRF, which the collector uses as the initial prior distribution. The d_i for all the irrelevant values are then set to 0.

Then, each parameter p_{ij} is specified based on a value drawn from the Beta distribution determined by d_i and d_j. The actual value of the p_{ij}'s are also unknown to the collector.

c. Example

The fixed relevance values d_j and the probabilities of relevant conversations p_{ij} in the graph shown in Figure 4 were drawn by the simulation, and the results are shown in Figure 5.

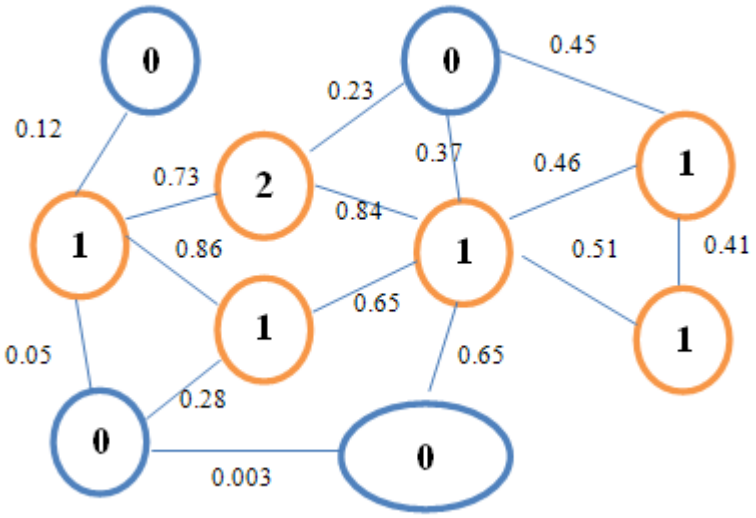


Figure 5. the graph and the probabilities (p_{ij})

5. Stage 4 – Screening a Conversation

Different algorithms (presented in Chapter III) are used to choose the sequence of edges from which conversations are screened. Once a conversation on an edge (i^*, j^*), still containing conversations, is chosen, the outcome of this conversation—relevant or irrelevant—is determined by drawing from a Bernoulli distribution with parameter $p_{i^*j^*}$. Then, the values of $n_{i^*j^*}, s_{i^*j^*}, f_{i^*j^*}$ are updated according to the outcome of the screening, the joint distribution of each pair (D_i, D_j) is updated accordingly (as shown in Chapter II), as well as the estimate of P_{ij}. The total number of relevant conversations (R) is updated at the end of each iteration.

6. Summary of the Simulation Parameters and Variables

a. Input Parameters Entered into the Simulation

Parameter	**Symbol**	**Stage**
Total number of nodes in the network	N	1
A graph of the relevant nodes ($d_i > 0$) in the network	G	1
Probability that two nodes will be connected	---	1
Mean number of conversations between two connected nodes	---	1
Scaling parameter to determine the joint distribution of the relevance values in each clique	p	2
Scaling parameter to determine the distribution of P_{ij} given the relevance values of i and j	q	2
The beta parameter for the Beta distribution	β	3
The probability that the relevance value of the node is identified after screening a conversation	c	4
Number of iterations of the simulation = number of conversations to be screened	T	4

Table 19. The input parameters entered into the simulation

b. Parameters Determined by the Simulation

Parameters	**Type**	**Symbol**	**Stage**
All the edges in the network	Linked List	---	1
Cliques in the network	Linked List	---	1
The function $\alpha(d_i, d_j)$, used for determining the alpha parameter for the Beta distribution	Table	$\alpha(d_i, d_j)$	2
True relevance value of node i	Integer	d_i	3
True probability that a conversation between i and j is relevant	Real Number	p_{ij}	3

Table 20. Parameters determined by the simulation

c. Variables Used in the Simulation

Variable	**Type**	**Symbol**	**Stage**
Number of unscreened conversations between nodes i and j	Integer	n_{ij}	1
List of Factors representing the cliques in the graph, the MRF used for the updating process	Factor List	$\{\psi(D_c)\}$	2
Number of relevant and irrelevant conversations screened between nodes i and j	Table	$\{s_{ij}\},\{f_{ij}\}$	2,4
The updated joint distribution of (D_i, D_j) for each edge (i, j)	Factor List	$\{\psi(D_i, D_j\}$	2
The expected probability that a conversation between nodes i and j is relevant following the screening of k conversations on that edge.	Array	$\{\mathrm{E}[S_{ij}^{(k)}]\}$	4
Total number of relevant conversations screened	Integer	R	4

Table 21. variables used in the simulation, i.e., parameters that change throughout the simulation run

7. Run Time Considerations

The run-time of the variable-elimination process might be very long. It is especially long when the graph is dense or when the average size of the cliques is relatively large. This affects the run-time of the algorithms, as after listening to a conversation, all the edge factors $\psi(D_i, D_j)$ need to be updated (unless the relevance values of both nodes are known).

There are alternative approximate algorithms to overcome this problem (Kohler at Friedman, 2010). However, we decided instead to use the variable elimination algorithm with two modifications:

- A partial updating of the network. After listening to a conversation between (i, j), we only update edges containing neighbors of i and j. The justification for this modification is that the change in the expected value of $P_{l,k}, l, k \neq i, j$ is usually very small. In addition, those other edges might be updated later on, when edges adjacent to them are chosen.

- A recursive use of the variable elimination algorithm. After screening a conversation, we need to update several joint distributions. If we perform the variable elimination algorithm sequentially, we would do unnecessary repetitions of calculations. We therefore use a recursive algorithm to avoid those repetitions.

B. THE ANALYSIS METHOD

1. Overview

In order to illustrate our model and examine the algorithms described in chapter III, we examine a case study. We construct a network whose topology is based on a terrorist organization in Tanzania (CSAOS, 2007). Due to lack of real-life data, we choose input parameters which would represent a plausible terrorist network, and would allow us to illustrate the performance of different algorithms. We then change some of the parameters and see how it affects the performance of those algorithms (shown in Chapter V).

2. The Network Graph

The network of 17 terrorists behind the 1998 United States embassy bombing in Tanzania, is depicted in Figure 6 (CSAOS, 2007).

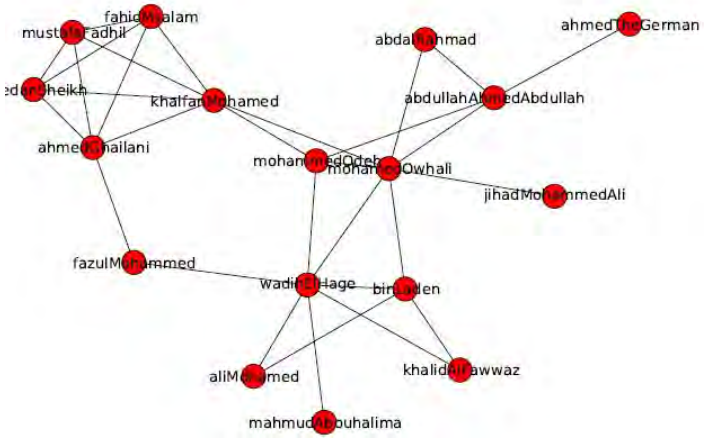


Figure 6. Network of the terrorists in charge of the U.S. embassy bombing in Tanzania

We added to this network 17 “dummy nodes,” representing people connected to the terrorists but not directly involved in the terrorist attack. As explained in section A.2, we added edges randomly among the dummy nodes and between them and the real nodes in the network. The resulting network is shown in Figure 7.

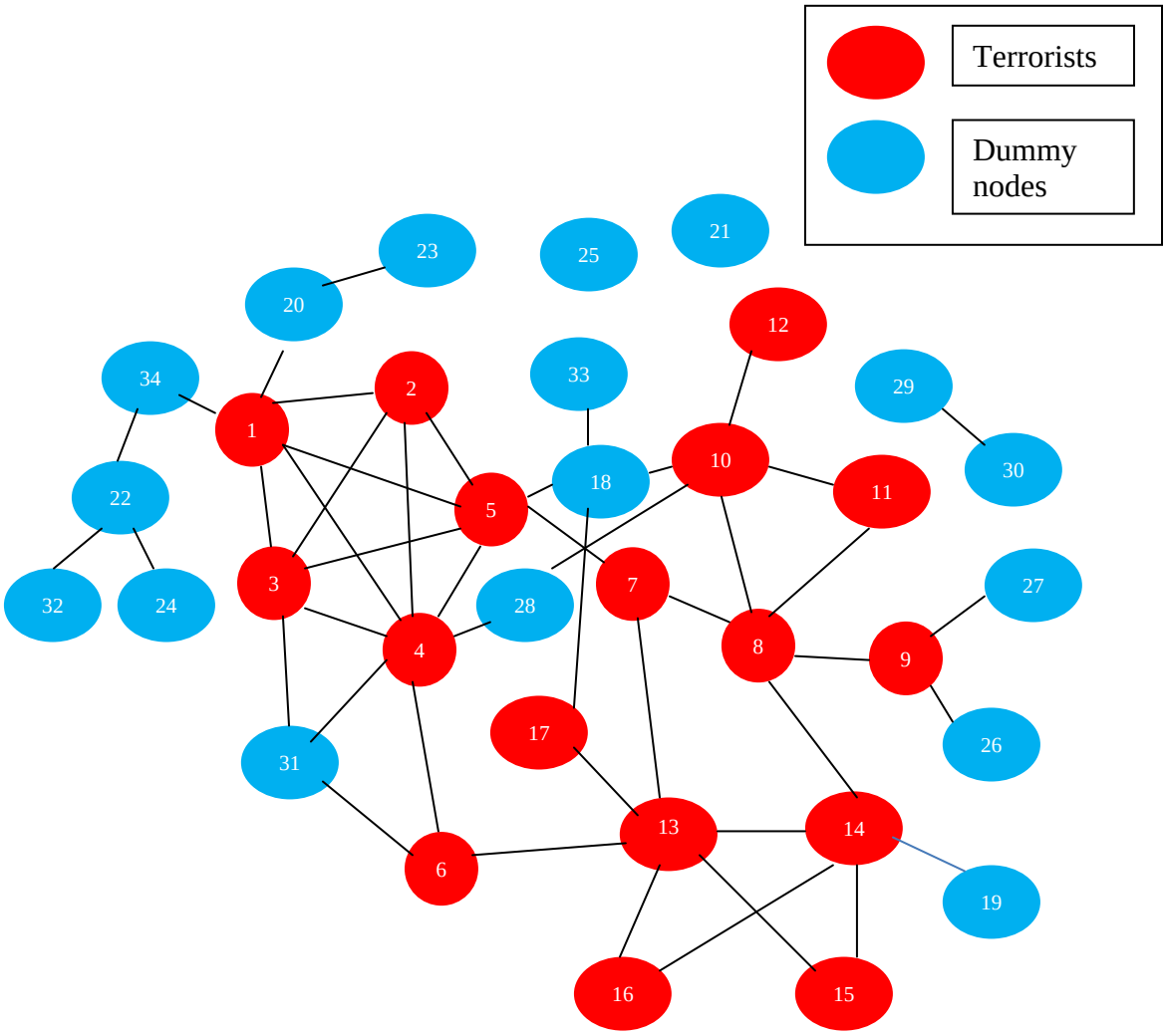


Figure 7. The network with dummy nodes

Red nodes represent the terrorists, and blue nodes are the randomly added dummy nodes. We chose the parameter $p = 0.05$ as the probability that a blue node is connected to any other node in the graph. As a result, some blue node are disconnected (21, 25),

some are connected to 3 or more nodes (18, 22, 31) and the others are connected to one or two other nodes. The choice of this parameter is rather arbitrary, and different values might have been chosen.

3. Case Study Parameters

The chosen input parameters for the simulation are shown in Table 22.

Parameter	**Symbol**	**Value**
Total number of nodes in the network	N	34
A graph of the relevant nodes ($d_i > 0$) in the network	G	Shown in Figure 6
Probability that two nodes are connected	---	0.05
Mean number of conversations between two connected nodes	---	100
Scaling parameter to determine the joint distribution of the relevance values in each clique	p	1
Scaling parameter to determine the distribution of P_{ij} given the relevance values of i and j	q	3
The beta parameter for the Beta distribution	β	1
Number of iterations for the simulation / number of conversations to be screened	T	300

Table 22. Parameters values for the case study

The number of iterations ($T = 300$) is a compromise between the run time and the ability to differentiate the different algorithms. With a lower number of iterations, the run time of the simulation is shorter. On the other hand, the higher the number of iterations, the easier it is to differentiate between the different algorithms. We therefore chose the value of 300 as an appropriate compromise. Then, the mean number of conversations (100) was chosen so that the simulation would illustrate how the algorithms handle the possibility that an edge would have no more conversations to screen. In Chapter V we change the mean number of conversations and examine how the results change.

The choice of the probability that two nodes are connected (0.05) and the total number of nodes in the network ($N = 34$) are limited by the requirement for a reasonable

run time of the simulation. For higher value of these parameters, the run time of the variable elimination algorithm would be much longer. Some methods to overcome this obstacle are mentioned in Chapter VI.

The value of p, the scaling parameter ($p = 1$) is set to represent homophility in the network while maintain some level of randomness for the relevance values. The scaling parameter q is set to be 3, so there would be a strong correlation between the relevance values d_i, d_j and the respective parameter p_{ij}.

The value of beta ($\beta = 1$) determines that there are only a few edges with a high value of p_{ij} (over 0.65) while the values of the other p_{ij} is significantly lower. The results given different values of the beta parameter are shown in chapter V.

As a result, the "state of the world" of the case study, i.e., the values of n_{ij}, d_i and p_{ij}, is shown in Table 23.

Node	1	2	3	4	5	6	7	8	9	10	11	12
d_i	1	2	2	2	1	2	1	1	1	1	1	1
Node	13	14	15	16	17	18	19	20	21	22	23	24
d_i	1	2	1	1	0	0	0	0	0	0	0	0
Node	25	26	27	28	29	30	31	32	33	34		
d_i	0	0	0	0	0	0	0	0	0	0		

Edge	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)
n_{ij}	98	100	107	122	113	98	110	104	96
p_{ij}	0.85	0.64	0.31	0.04	0.51	0.20	0.51	0.91	0.03
Edge	(4,5)	(4,6)	(5,7)	(5,8)	(6,13)	(7,8)	(7,13)	(7,10)	(8,10)
n_{ij}	91	109	97	100	117	120	101	99	98

p_{ij}	0.0005	0.58	0.006	0	0.11	0.16	0.0001	0.005	0.69
Edge	(8,9)	(8,11)	(8,13)	(8,14)	(10,11)	(10,12)	(13,14)	(13,15)	(13,16)
n_{ij}	103	92	102	97	93	99	102	100	107
p_{ij}	0.071	0.06	0.0002	0.03	0.0007	0.001	0.8	0.08	0.27
Edge	(14,15)	(14,16)	(18,33)	(18,17)	(18,5)	(18,10)	(19,14)	(20,23)	(20,1)
n_{ij}	104	102	97	98	94	97	105	101	105
p_{ij}	0.58	0.28	0	0.004	0.007	0.18	0.16	0	0.0001
Edge	(22,24)	(22,32)	(26,9)	(27,3)	(27,9)	(28,4)	(28,10)	(29,30)	(31,3)
n_{ij}	90	99	103	91	103	113	99	101	100
p_{ij}	0	0	0	0.22	0.05	0.076	0.004	0	0.21
Edge	(31,4)	(31,6)	(34,22)	(34,1)					
n_{ij}	100	91	99	100					
p_{ij}	0.58	0.064	0.003	0.003					

Table 23. The values of n_{ij} and p_{ij} (i.e., the number of conversations in each edge and the probability that a conversation is relevant)

The graph in Figure 8 shows the network, where the thickness of the edges represent the likelihood of a relevant conversation.

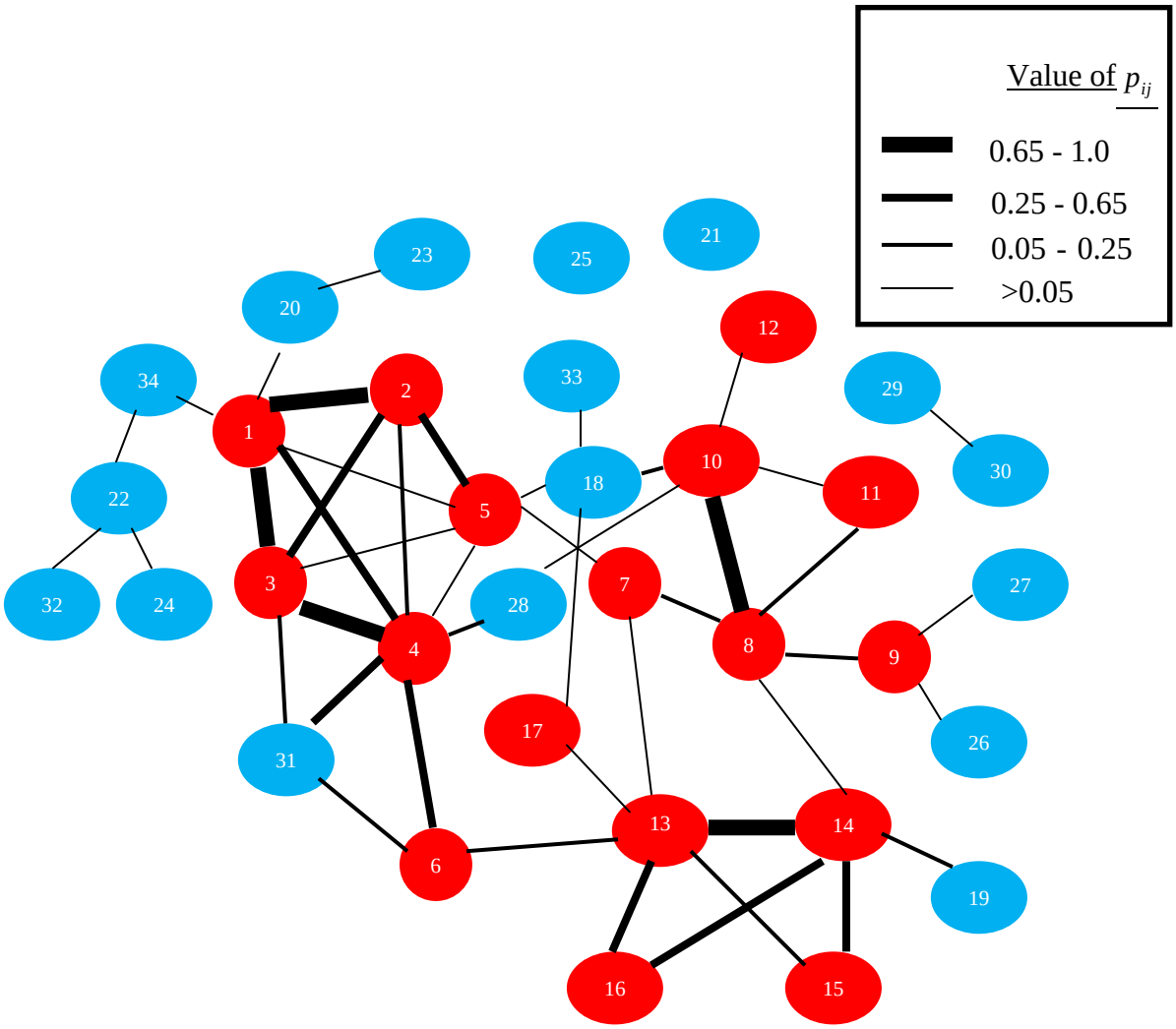


Figure 8. The values of the p_{ij}

C. THE ALGORITHMS STUDIED AND COMPARED

1. The Algorithms

Utilizing the "Tanzania" case study, we analyze and compare five algorithms described and discussed in Chapter III.

- *Pure Exploitation* (PE): A greedy algorithm which chooses each iteration the conversation with the highest likelihood to be relevant.
- *Softmax*: An algorithm which assigns each edge with a weight according to the probability that a conversation from that edge is relevant, and chooses randomly based on those weights. The algorithm is described in Chapter III.
- *Modified VDBE (VDBE)*: This algorithm is based on the ε-greedy Value-Difference-Based-Exploration algorithm, described in Chapter III. According to the original algorithm, each iteration the collector chooses whether to explore or to exploit. When the collector explores, he chooses a random edge. When he exploits, he chooses an edge the set of edges with the highest value of $\mathrm{E}[P_{ij}]$. The probability to explore is ε, where the value of ε is updated throughout the screening process in response to the results. The rate in which epsilon changes depends on the differences in the values of $\mathrm{E}[P_{ij}]$ from iteration to iteration: the higher the changes in $\mathrm{E}[P_{ij}]$, the lower the decay rate of epsilon, and the collector will more likely choose to explore.

  This algorithm was originally designed to choose between uncorrelated alternatives. When the alternatives are correlated, the random exploration proves ineffective, as it ignores the collector's assessment regarding alternatives which have not been examined yet. We therefore modified the original algorithm, and instead of a random exploration use the Softmax Algorithm when the collector chooses to explore. The parameter of the Softmax algorithm (the temperature) is relatively high (0.25), so the collector would tend to explore different alternatives.

- *Wide-Exploration-First* (WEF): This algorithm combines the Exploration-First heuristic and the wide exploration method, both mentioned in chapter III. According to this algorithm, during the first iterations (the exploration period) the edges are chosen according to the wide exploration heuristic (explained in Chapter III) – choosing different edges such that each edge is sampled less than a predetermined number of times. Then, during the exploitation period, edges are chosen according to the Softmax algorithm whose parameter is relatively low (0.05), so the collector would prefer exploitation over exploration.

- *Knowledge-Gradient-Exploration-First* (KGEF): This algorithm is similar to the WEF algorithm, except for a different exploration policy. In this algorithm, during the exploration period the edges are chosen according to the Knowledge Gradient (KG) policy. According to this policy, each round the collector chooses an edge which is most likely to change his assessment of which edges should he choose during the next rounds.

2. Choosing the Parameters for the Algorithms

In order to determine the optimal values of the parameters for an algorithm, we can try a variety of different values until the optimal values are found (e.g. [Tokic, 2010]). Instead, we only examine several possible values for each parameter, and thus have a rough estimation of what the optimal value of each parameter is. We believe that this method of choosing the parameters is sufficient considering the desired level of accuracy.

- Pure Exploitation: No parameters.

- Softmax: The algorithm has one parameter, called *the temperature*. The temperature parameter determines how much the collector focuses on high value alternatives (i.e., edges with a high value of $\mathrm{E}[P_{ij}]$).

The weights assigned to each edge are $w_{ij} = \frac{e^{\frac{v_{ij}}{T}}}{\sum_{(i,j)} e^{\frac{v_{ij}}{T}}}$ where T is the temperature and $v_{ij} = \mathrm{E}[P_{ij}]$. Therefore, to estimate the desired value of T, one can examine the ratio $\frac{e^{\frac{v_{high}}{T}}}{e^{\frac{v_{ave}}{T}}}$ where v_{high} is a typical high value of $\mathrm{E}[P_{ij}]$ (in our case study, it is about 0.8) and v_{ave} is the average expected value of the unscreened node (in our case study, about 0.2). The graph in Figure 9 shows how the temperature parameter affects this ratio. For example, when $T = 0.05$ it will almost always choose edges with higher $\mathrm{E}[P_{ij}]$. When $T = 0.25$, it is more likely to choose edges with a high $\mathrm{E}[P_{ij}]$, but is still likely to choose other edges as well.

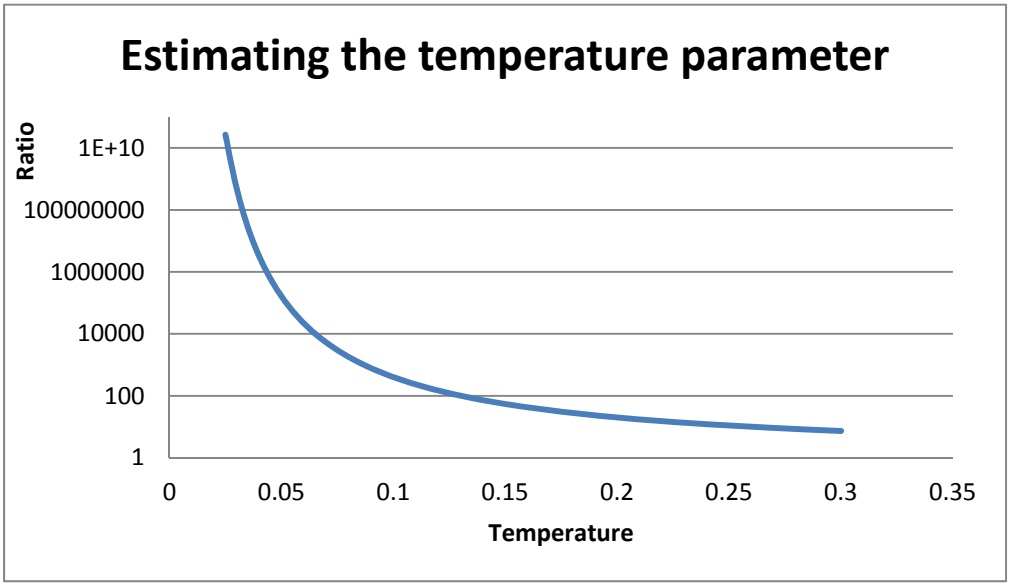


Figure 9. The effect of the temperature parameter in the Softmax algorithm on the ratio between weight of a high-value edge and average-value edge

Based on the graph in Figure 9, we compared several values for the temperature parameter (0.05, 0.08, 0.1, 0.12, 0.15) and 0.08 is proved to be the optimal choice. Interestingly, in this region even slight changes in the parameter (of about 0.02–0.03) have a significant effect on the outcome (a difference of 10–20 screened conversations).

- Modified VDBE: The modified VDBE algorithm has four parameters. Epsilon is determined by the expression $\varepsilon^{(k+1)} = \delta \frac{1-e^{\frac{-\left|v_i^{(k+1)}-v_i^{(k)}\right|}{\sigma}}}{1+e^{\frac{-\left|v_i^{(k+1)}-v_i^{(k)}\right|}{\sigma}}} + (1-\delta)\varepsilon^{(k)}$ (equation 3.4) and therefore depends on the parameters δ and σ. For the exploitation criterion, the algorithm chooses randomly between a set of edges with the highest value of $\mathrm{E}[P_{ij}]$. The size of this set is the third parameter. For the exploration, we need a temperature parameter for the Softmax algorithm.

  The set size for the exploitation is likely to be a small integer, and setting the size to 1 (i.e., always choosing the best alternative) proved to provide the best results. According to Figure 14, we chose the temperature of the Softmax to be 0.25, to ensure that different edges are chosen.

  Choosing δ and σ proved to be relatively complicated. Both those parameters determine the decay rate of epsilon. The δ parameter is the decay rate of epsilon given that the system is stable, i.e., when there are very few changes in the values of the $\mathrm{E}[P_{ij}]$. This is in a way an upper bound on the actual decay rate. The σ parameter (called the sensitivity parameter [Tokic, 2010]) determines how much changes in the values of $\mathrm{E}[P_{ij}]$ reduce the total decay rate. Since the typical changes of the values before reaching a stable state can be estimated (in our case study they are about 0.1–0.15), we can estimate the expected decay with and without changes in the values of $\mathrm{E}[P_{ij}]$. For example, for $\delta = 0.02$ and $\sigma = 0.3$, the decay of epsilon when the values do not change and when they do change is shown in Figure 10.

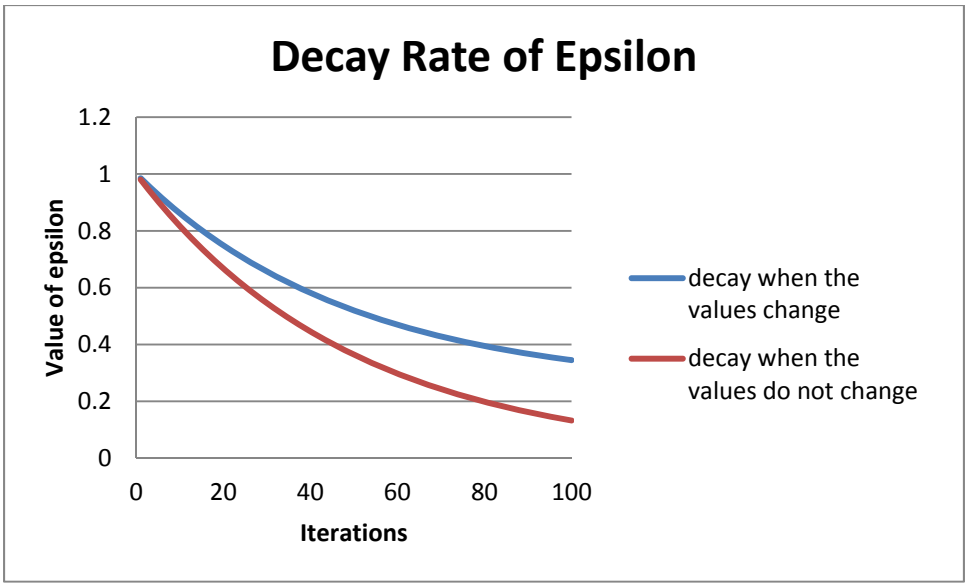


Figure 10. The decay rate of epsilon when the values of $\mathrm{E}[P_{ij}]$ change and when they remain about the same, given that $\delta = 0.02$ and $\sigma = 0.3$.

Tokic suggests that δ would be set to be one over the number of alternatives [Tokic, 2010]. However, this number proved to be too low. By changing the parameters, we can actually set the algorithm to be pure exploitation or Softmax. When $\delta \sim 1$ and σ is very large, the decay rate is very high, and therefore the algorithm would almost immediately starts exploitation, as in the PE algorithm. When δ is close to zero, the value of epsilon remains constant, and think the initial value of epsilon is one, the algorithm would always choose to explore. Since during the exploration iterations the algorithm chooses the edges according to the Softmax algorithm, the algorithm is then effectively equivalent to Softmax. To avoid these possibilities, we limited δ to be between 0.02 and 0.2. After examining different values for δ (0.02, 0.06, 0.1, 0.15) and for σ (0.1, 0.2, 0.3, 0.4, 0.5), the optimal values proved to be $\delta = 1$ and $\sigma = 0.4$.

- Wide Exploration First: This algorithm has three parameters: the number of exploration rounds at the beginning, the maximal number of conversations to be screened from the same edge during the exploration phase (β), and the temperature parameter for the Softmax algorithm during the exploitation phase.

  The setting of the β parameter is derived from the tradeoff between exploring as many edges as possible, and being able to determine which edges are better. The higher β is the more conversations are screened from the edges. Therefore, if β is high, at the end of the exploration phase the collector has a better assessment of the true values of p_{ij} for the edges he has sampled.

  The temperature parameter is set to be 0.05, to ensure the choice of edges with a high value of $\mathrm{E}[P_{ij}]$. We examined several options for the number of exploration rounds (20, 30, 40) and the maximal number of conversations (1, 2, 3) and 20 exploration rounds with up to three conversations from each edge seemed to be the optimal choice.

  In a real life scenario, the collector can simply choose along the way when to stop exploring and start exploiting, based on the results so far.

- KG-Exploration First: This algorithm has two parameters, the number of exploration rounds and the temperature parameter for the Softmax algorithm. As with the WEF algorithm, the temperature parameter is set to be 0.05. After examining several choices for the number of exploration rounds (20, 30, 40, 50), the value of 40 seems to be the optimal choice.

3. The Parameters Values

The values of the parameters for the algorithms are summarized in Table 24.

The algorithm:	**The parameters:**
Pure Exploitation	None
Softmax	Temperature – 0.08
KG-VDBE	δ - 0.1 σ - 0.4 # top edges form which the algorithm chooses to exploit – 1
WEF	# rounds of exploration (i.e length of the exploration stage) – 20 # of samples from each edge (B) - 3 Temperature for exploitation stage – 0.05
KGEF	# rounds of exploration (i.e length of the exploration stage) – 40 Temperature for exploitation stage – 0.05

Table 24. The chosen values of the parameters for each algorithm

V. RESULTS

A. ALGORITHMS ILLUSTRATION

Before analyzing the overall performance of the different algorithms mentioned in Chapter IV, we illustrate the behavior of each algorithm based on a single run of the simulation described in Chapter IV. The single run is chosen randomly. For each algorithm, we examine in each iteration (i.e., selection of an edge in the network) the accumulated number of relevant conversations the algorithm has already found. In addition, we examine in each iteration the difference $\max_{(i,j)}\{p_{ij}\} - p_{i^*j^*}$, where (i^*, j^*) is the chosen edge. In other words, we examine the distance between the true p_{ij} value of the chosen edge and the edge with the highest value of p_{ij} among those edges whose conversations have not been exhausted.

1. Pure Exploitation (PE)

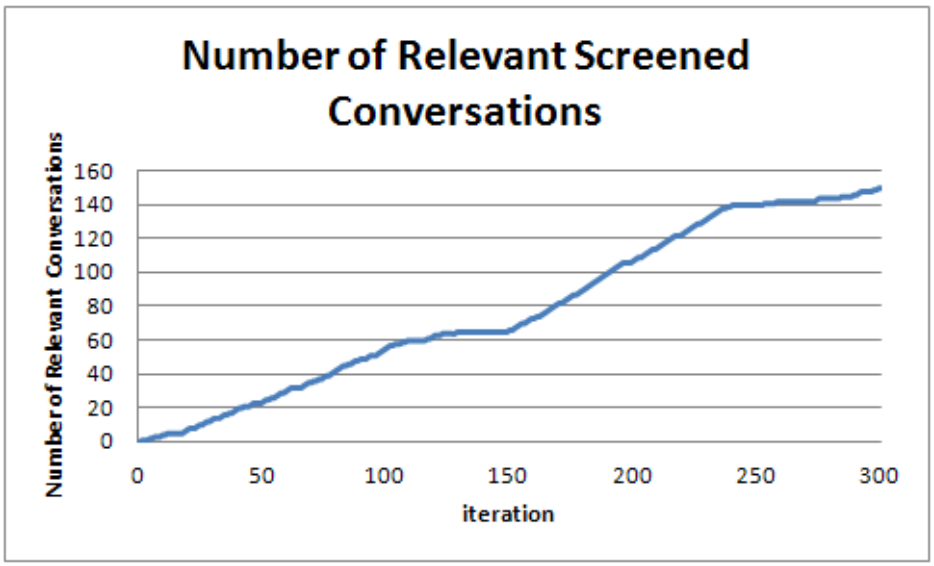


Figure 11. The accumulated number of relevant conversations, based on a single run of the PE algorithm.

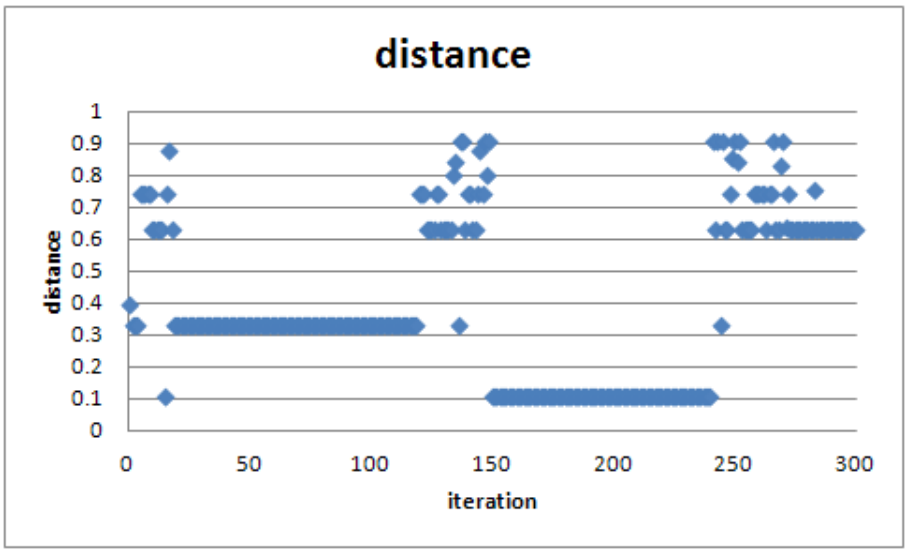


Figure 12. The distance between the p_{ij} of the chosen edge and the maximal possible at each iteration, based on a single run of the PE algorithm.

The PE algorithm is a simple greedy algorithm, which chooses each iteration the edge with the highest expected probability to produce a relevant conversation. Figure 12 shows how the algorithm spends several iterations sampling different edges, and then focuses on a single edge. After about 100 iterations, when there are no more conversations to be screened from an edge, the algorithm starts searching again. The chosen edge can be very close to the optimal, as between the 150th and 250th iterations, or sub-optimal (i.e., an edge whose value of p_{ij} is significantly lower than the maximal possible) as between the 20th and 120th iterations and during the last 50 iterations.

2. Softmax

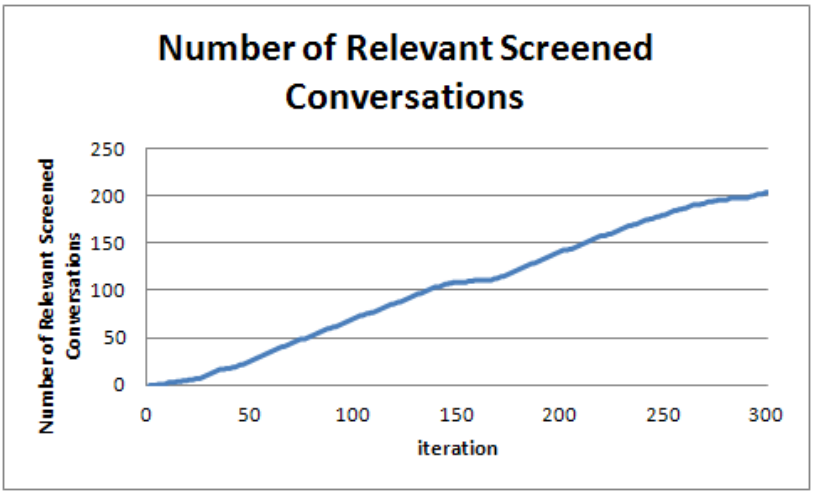


Figure 13. The accumulated number of relevant conversations, based on a single run of the Softmax algorithm.

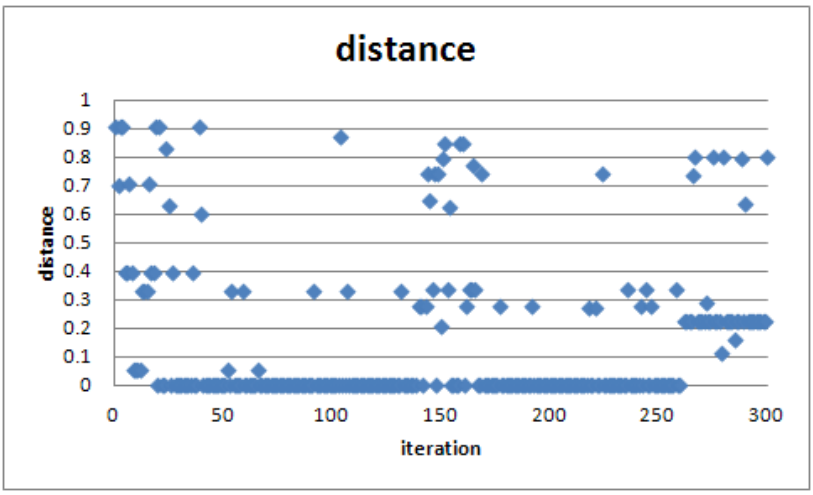


Figure 14. The distance between the p_{ij} of the chosen edge and the maximal possible at each iteration, based on a single run of the Softmax algorithm.

Similarly to the PE algorithm, the Softmax algorithm spends several iterations examining different edges, and then focuses on a specific edge. Specifically, based on Figure 13, between the 30th and 140th iterations, the algorithm focuses on the optimal edge (distance 0), after it runs out of conversations it spends a few iterations searching, and then chooses again the optimal edge. However, unlike the PE algorithm, even after

focusing on one edge the algorithm occasionally examines other edges. The rate in which other edges are examined depends on the value of p_{ij} for the chosen edge: during the last 40 iterations when the chosen edge is sub-optimal (i.e., the p_{ij} of the chosen edge is significantly lower than the maximal possible), the algorithm examines other edges more often.

3. VDBE

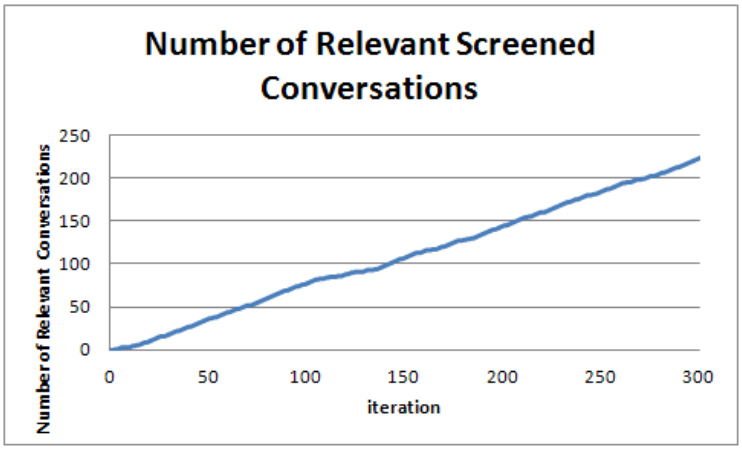


Figure 15. The accumulated number of relevant conversations, based on a single run of the VDBE algorithm.

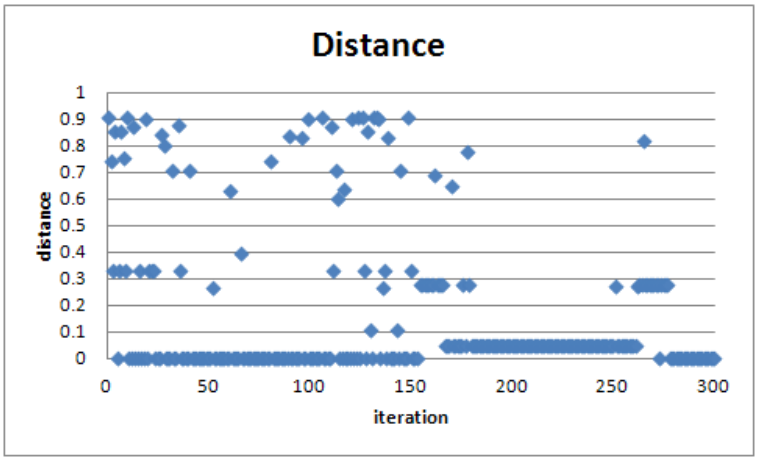


Figure 16. The distance between the p_{ij} of the chosen edge and the maximal possible at each iteration, based on a single run of the VDBE algorithm.

As can be seen in Figure 16, the VDBE algorithm alternates between focusing on one edge (exploitation) and exploring different edges (explorations), as the Softmax algorithm. However, the rate in which the VDBE algorithm explores other edges greatly depends on the number of iterations, and as that number increases the algorithm almost only exploits. As the distance increases from the 270th rounds onward (i.e., the chosen edge is sub-optimal), the algorithm starts occasionally exploring. The exploration leads the algorithm to divert from the sub-optimal edge it chooses and focus on the optimal edge during the last 30 iterations.

4. KGEF

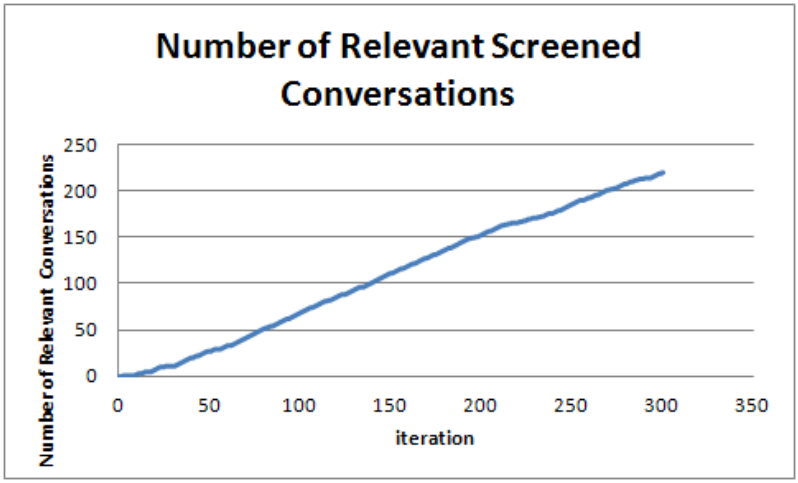


Figure 17. The accumulated number of relevant conversations, based on a single run of the KGEF algorithm.

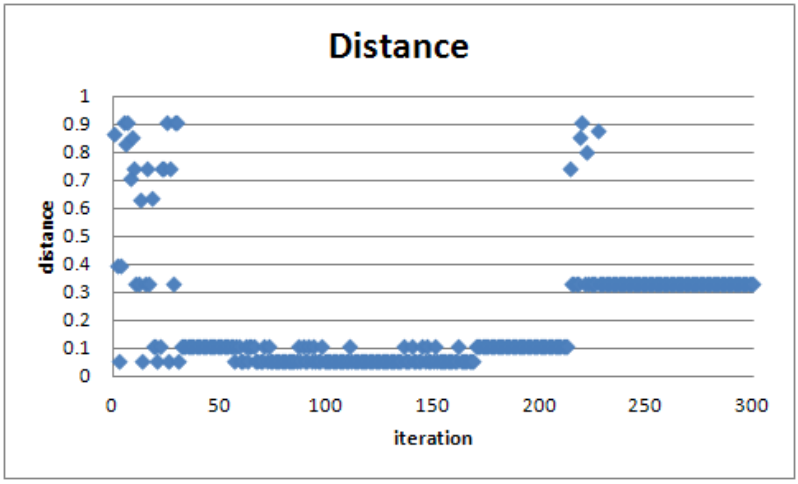


Figure 18. The distance between the p_{ij} of the chosen edge and the maximal possible at each iteration, based on a single run of the KGEF algorithm.

Based on Figure 18, the GK algorithm spends the first iterations exploring different edges. Then, for 200 iterations its performance is really close to optimal. However, After 200 iterations the algorithm starts searching for other edges, and eventually focuses on a sub-optimal edge, which explains why after 200 iterations the curve in Figure 17 increases in a much slower rate. Although this is not always the case, many times the algorithm indeed focuses initially on an edge which is close to the optimal one, but later focuses on a sub-optimal edge. A possible explanation is that the KG policy provides one or two edges which are very close to the optimal value, but does not show which the next best edges are.

5. WEF

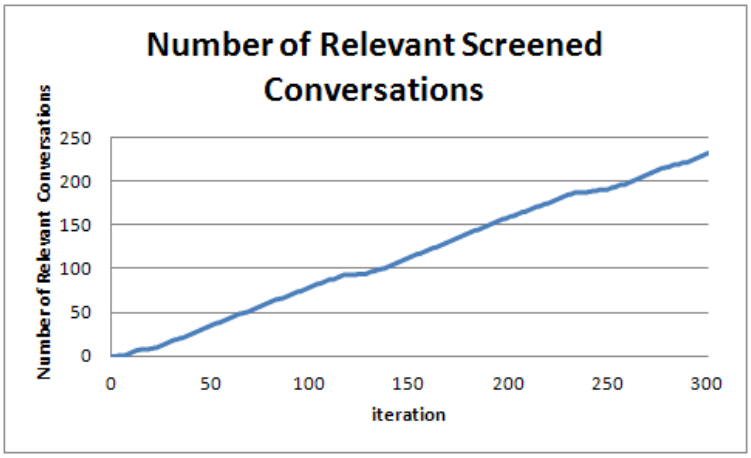


Figure 19. The accumulated number of relevant conversations, based on a single run of the WEF algorithm.

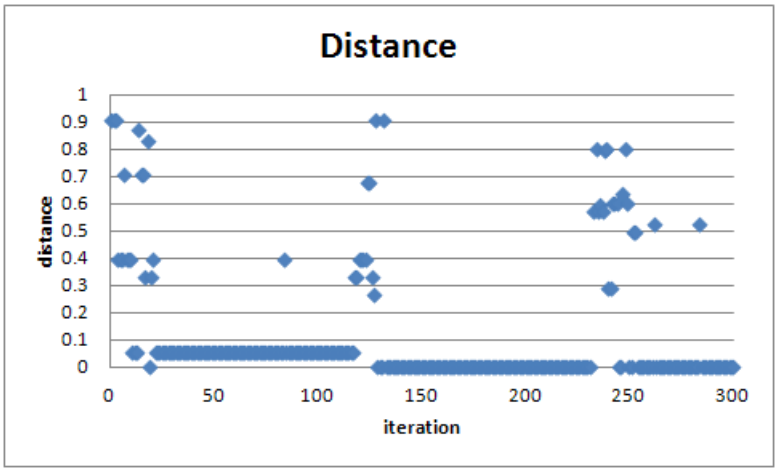


Figure 20. The distance between the p_{ij} of the chosen edge and the maximal possible at each iteration, based on a single run of the KGEF algorithm.

Based on Figure 20, during the first 20 iterations, the WEF algorithm explores different edges. Then, during the exploitation stage, it focuses on a relatively close to optimal edge until it runs out of conversations (after about 100 iterations), samples different edges for several iterations and then focuses on another edge.

B. CASE STUDY RESULTS

The following analysis of the case study presented in Chapter IV is based on 150 runs of each of the five algorithms (presented in the previous section).

1. Overall Comparison

Figure 21 shows the average number of relevant conversations detected by each algorithm. The algorithms are compared to a so called "perfect" algorithm, in which the p_{ij} are known, and at each iteration the edge with the highest value of p_{ij} is chosen. The error bars are calculated according to the 95% confidence interval $[\overline{x} - z_{0.025}\frac{s}{\sqrt{n}}, \overline{x} + z_{0.025}\frac{s}{\sqrt{n}}]$, where $\overline{x}$ is the sample mean, $z_{0.025}$ is a constant derived from the standard normal distribution and equals 1.96, s is the standard deviation of the sample (shown in Figure 22) and n is the sample size (in our case, $n = 150$) (Devore, 2009).

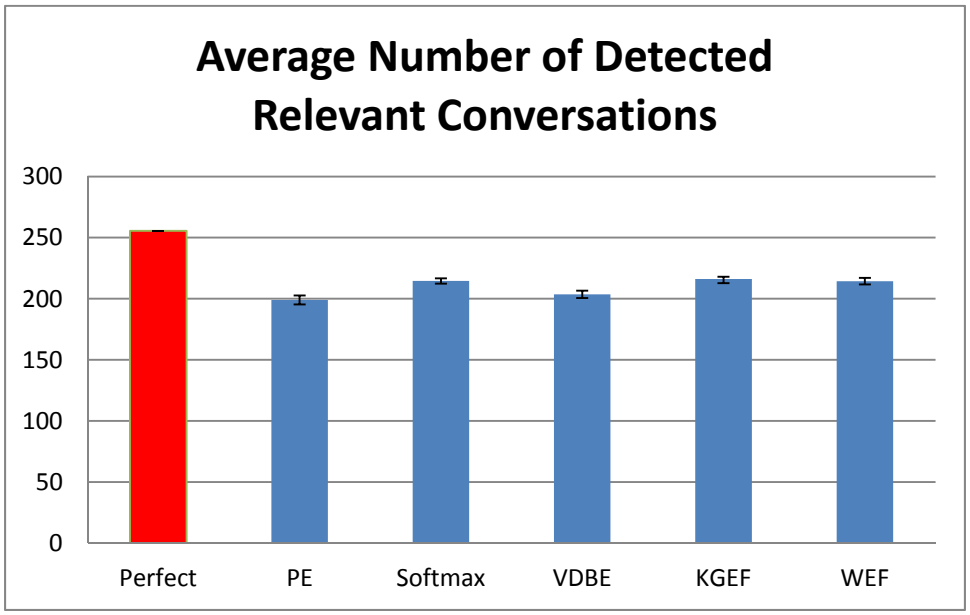


Figure 21. Average number of relevant conversations after 300 iterations, based on 150 runs of each algorithm.

It seems that although the PE algorithm has the worst performance, it still performs relatively well, as the difference between it and the algorithms is not very large (about 15 relevant conversations, less than 10% of the number of conversations). Figure 21 shows that the performance of the VDBE algorithm is worse than the performance of the Softmax, WEF and KGEF algorithms. There is no clear distinction (given 95% confidence) between the VDBE and the PE algorithms. As for the other three algorithms, Softmax, WEF and KGEF, there is no clear distinction between their performances (given 95% confidence).

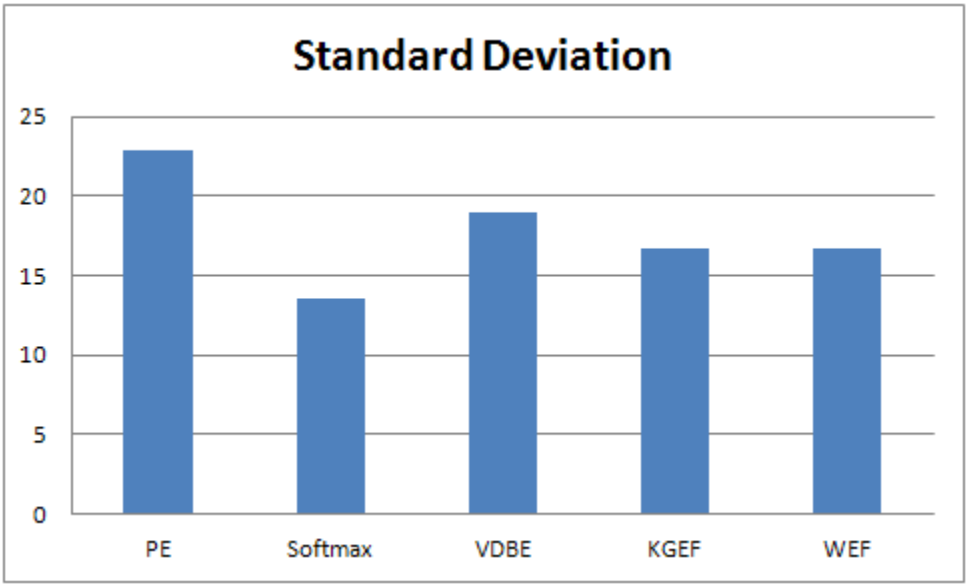


Figure 22. The standard deviation of each algorithm

Although the difference in the average number of conversations is relatively small, the difference in the standard deviation is much higher (as seen in Figure 22). The PE algorithm has the largest standard deviation, which is expected as the PE algorithm tends to focus relatively quickly on one edge until there are no more conversations to screen from this edge. Therefore, the performance of the PE algorithm greatly depends on whether the choice of the edge is optimal or sub-optimal. Softmax has the smallest standard deviation.

2. The Behavior of each Algorithm

We can gain insights regarding the behavior of the algorithms by examining not only the final outcome, but their performance throughout the process. For each iteration k, we take the average over all the runs of the difference $R^{(k+1)} - R^{(k)}$, where $R^{(k)}$ is the number of relevant conversations accumulated by the kth iteration. That average is a number between 0 and 1, and represents the rate in which relevant conversations are accumulated in the kth iteration. The average is denoted by $\overline{p}^{(k)}$, as for a very large number of runs it would equal the average over the p_{ij} of the chosen edge in each iteration. We will compare the algorithms by examining the value of $\overline{p}^{(k)}$ in each iteration. For convenience, we separate the comparison into two. Using the Perfect and PE algorithms as a baseline, we compare Softmax with VDBE (Figure 23), and WEF with KGEF (Figure 24).

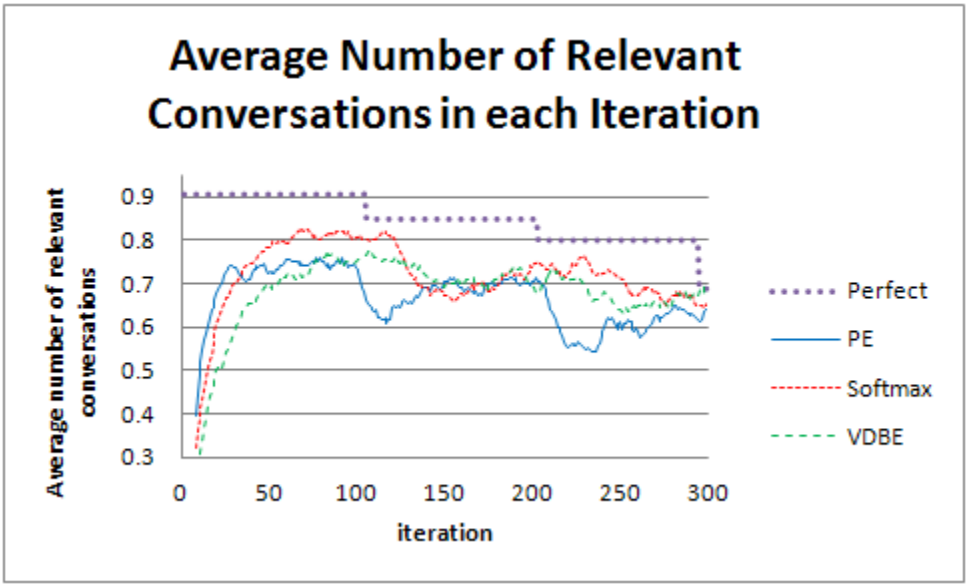


Figure 23. The average number of relevant conversations in each iteration for the PE, Softmax and VDBE algorithms.

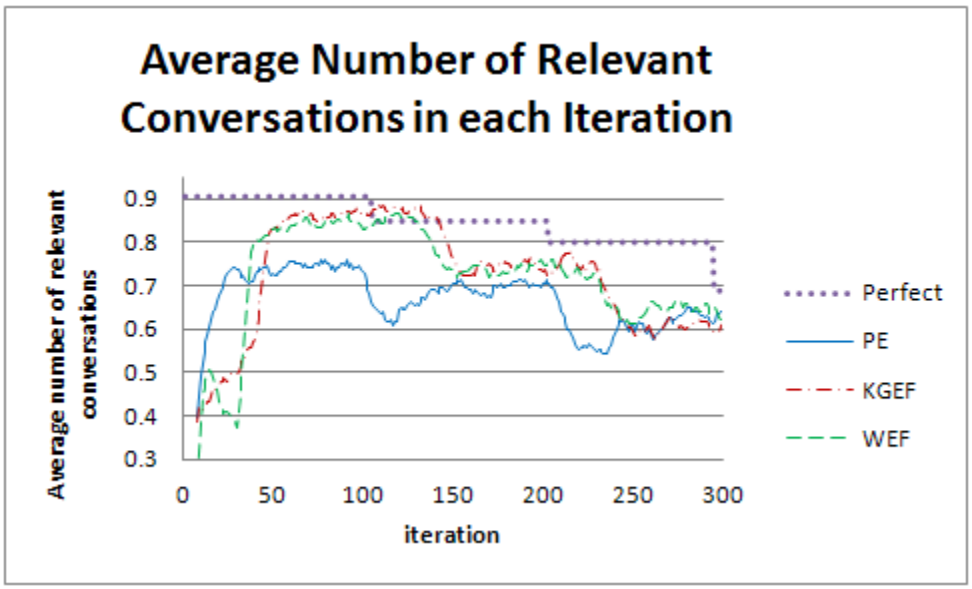


Figure 24. The average number of relevant conversations in each iteration for the PE, KGEF and WEF algorithms.

a. The PE Algorithm

Figures 23 and 24 show that during the first 20 iterations the PE algorithm has a higher value of $\overline{p}^{(k)}$, but as this value remains constant until the 100th iteration, the value of $\overline{p}^{(k)}$ for the other algorithms increases and all of them but the VDBE surpass it. It seems that the PE algorithm tends to focus relatively quickly on a single edge, but this edge might be sub-optimal (i.e., with a relatively low value of p_{ij}). The other algorithms require more time before focusing on a single edge, but then they tend to focus on an edge with a higher value of p_{ij}. Around the 100th and 200th iterations, the value of $\overline{p}^{(k)}$ drops abruptly, and stabilizes again after about 10–20 iterations. The reason is probably that the mean number of conversations to be screened from each edge is 100, and the drop happens after an edge runs out of conversations.

b. The Softmax Algorithm

Figure 23 shows that the value of $\overline{p}^{(k)}$ for this algorithm increases during the first 50 rounds. The value is then relatively high compared to PE and VDBE, although it is less than the value for WEF and KGEF. From that point onward, the value

gradually decreases, with a relatively steep descent after 100 iterations, probably for the same reason mentioned above for the PE algorithm (an edge ran out of conversations).

c. The VDBE Algorithm

According to Figure 23, the algorithm has a similar behavior to the Softmax algorithm: the value of $\overline{p}^{(k)}$ gradually increases during the first 50–70 iterations, and then gradually decreases. Compared to the Softmax algorithm, the value of $\overline{p}^{(k)}$ after 50–70 iterations is relatively low and similar to that of the PE algorithm. However, its decrease is much more gradual, and therefore that difference becomes less and less significant from the 120th iteration onward.

d. The KGEF Algorithm

Figure 24 shows that the value of $\overline{p}^{(k)}$ for the KGEF algorithm gradually increases during the first 40 iterations, until it reaches a very high value (very close to 0.9, the p_{ij} with highest value) at the end of the exploitation iterations of the algorithm. However, the value of $\overline{p}^{(k)}$ decreases at the 120th and 220th iterations, and during the last 80 iterations reaches about the same value as the $\overline{p}^{(k)}$ of the PE algorithm.

e. The WEF Algorithm

Figure 24 shows that the WEF algorithm has a similar behavior to the KGEF algorithm, as both has a relatively low value of $\overline{p}^{(k)}$ during the first 30–50 iterations, then the value becomes relatively high (close to 0.9), and gradually decreases. The difference between KGEF and WEF is that during the first 30 iterations, while the value of $\overline{p}^{(k)}$ for the KGEF gradually increases as the algorithm focuses on the optimal choices, the value of $\overline{p}^{(k)}$ for the WEF algorithm remains relatively low, as the algorithm explores more and more different edges. The WEF algorithm compensates on that, as the decrease of $\overline{p}^{(k)}$ is slower. As can be seen in Figure 24, between the 200th and 250th iteration the performance of the WEF algorithm is better than that of the KGEF algorithm.

C. CHANGING THE SIMULATION PARAMETERS

We now describe how changing some of the simulation parameters mentioned in Chapter IV affects the performance of the different algorithms.

1. Mean Number of Conversations

The original mean number of conversations in each edge (n_{ij}) is 100. We now show how changing this number while keeping the other values constant affects the performance of the algorithm. To do that, we checked what happens when the mean number is reduced to 30, or increased to 350 (effectively meaning that each edge cannot be exhausted).

For each variation, we reexamined the chosen values of the algorithm parameters. As in Chapter IV, we selected several possible values for the parameters of each algorithm. We then compared the results given each possible value, and chose the parameters which resulted in the best results. The new chosen values are shown in Table 25.

The algorithm:	**The parameters:**	**Mean = 100**	**Mean = 350**	**Mean = 30**
Pure Exploitation	None			
Softmax	Temperature	0.08	0.08	0.05
KG-VDBE	δ	0.1	0.1	0.15
	σ	0.4	0.4	0.4
WEF	# iterations of exploration	20	30	20
	# of samples from each edge (B)	3	3	3
KGEF	# iterations of exploration	40	30	30

Table 25. The chosen parameters for the algorithms, given different mean number of conversations per edge.

There are very few changes when increasing the mean number to 350. However, when decreasing the mean number of conversations to 30, the parameters are changed as to prefer exploitation over exploration: The temperature in the Softmax parameter reduces from 0.08 to 0.05, and thus the algorithm tends towards focusing on edges with a high value of $\mathrm{E}[P_{ij}]$; The parameter δ in the VDBE algorithm increases from 0.1 to 0.15, and therefore the algorithm tends to exploit more (as explained in Chapter IV); The number of exploration iterations for the KGEF algorithm decreases from 40 to 30. The reason for that tendency is that exploration provides the collector with information about other edges in the network. This information, however, becomes less valuable when the mean number of relevant conversations decreases.

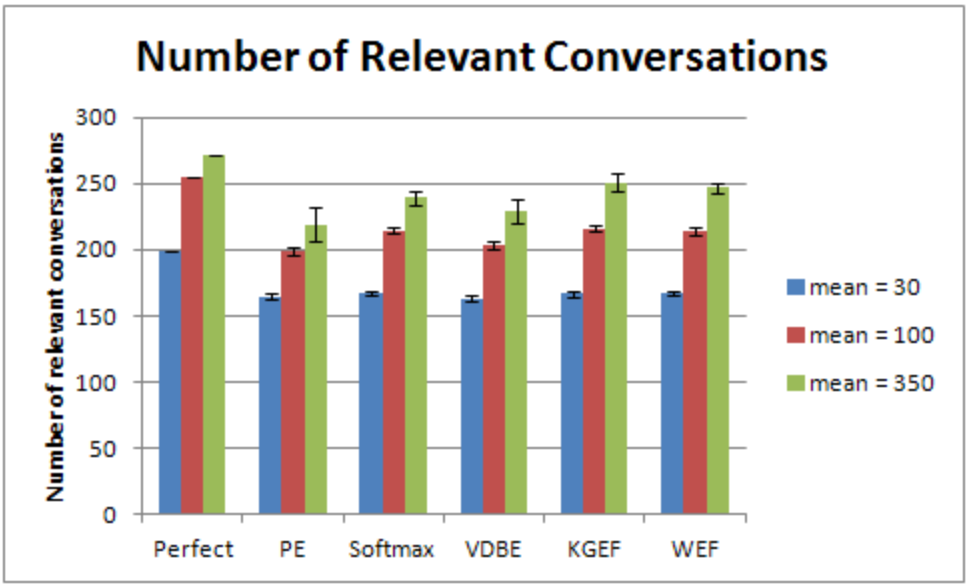


Figure 25. Accumulated number of detected relevant conversations after using each algorithm, given a different mean number of conversations; The distinction between the algorithm is better as the mean number of relevant conversations increases.

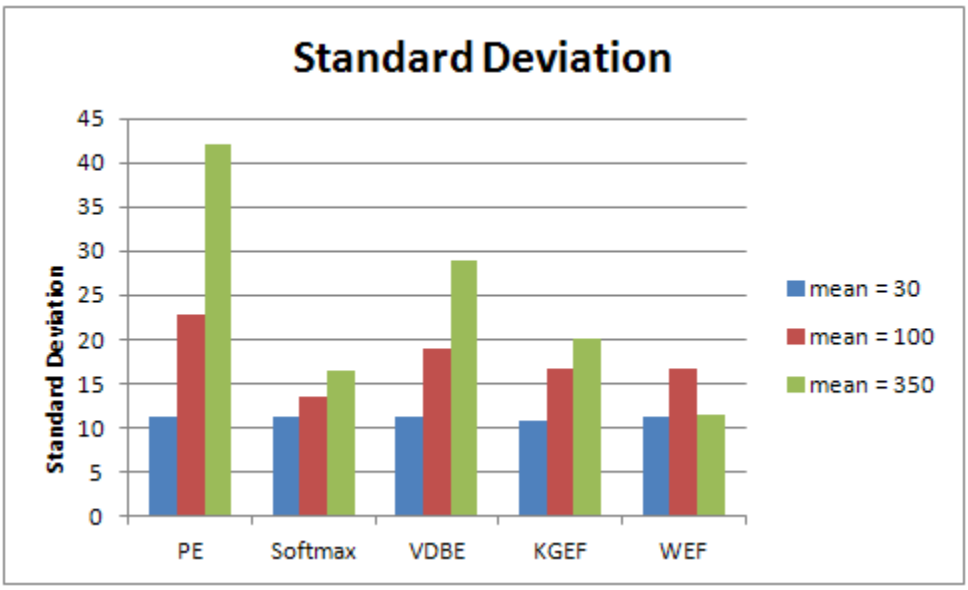


Figure 26. Standard deviation of the different algorithms, given different mean number of conversations; the difference between the algorithms is more significant as the mean number of relevant conversations increases.

The results are summarized in Figures 25 and 26. As expected, the higher the mean number of conversations, the better the results. When the mean number of conversations is 30, there are no significant differences (with 95% confidence) between the number of relevant conversations screened using each algorithm. In addition, the standard deviations of the algorithms are almost the same. When the mean is 350, the algorithms WEF and KGEF perform significantly better than PE, Softmax and VDBE. A more significant difference is with the standard deviations: PE has a very large standard deviation, then VDBE, followed by KGEF, Softmax and WEF. The WEF algorithm maintains a relatively low standard deviation.

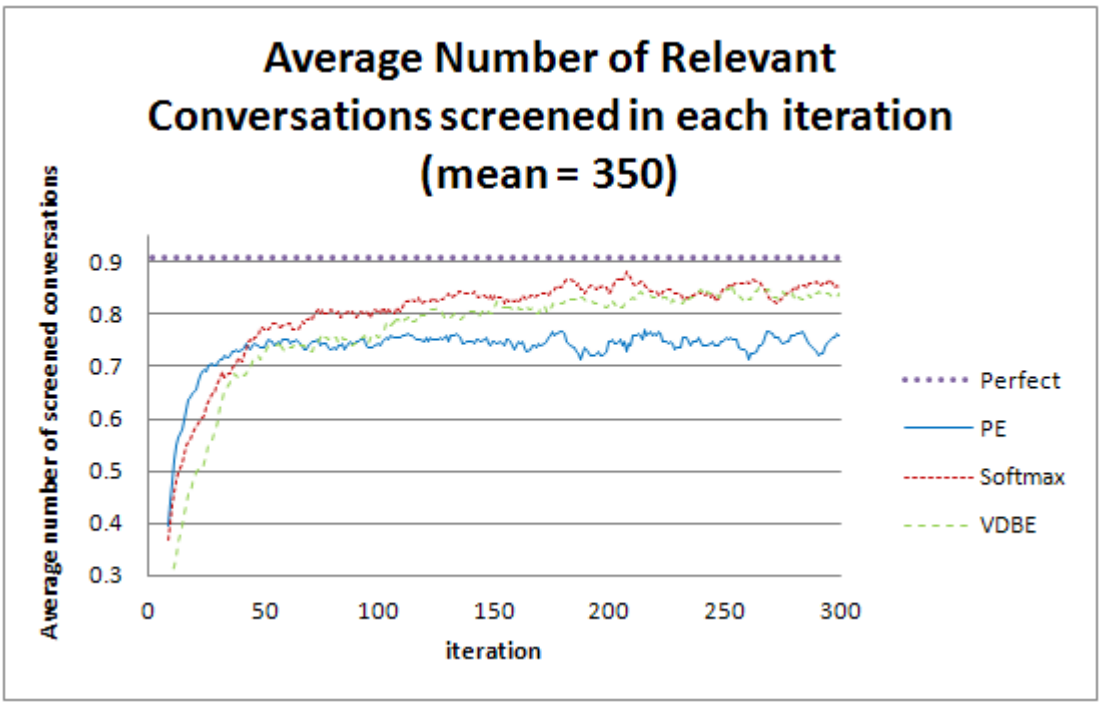


Figure 27. The value of $\overline{p}^{(k)}$ for the algorithms PE, Softmax and VDBE with a mean of 350 conversations.

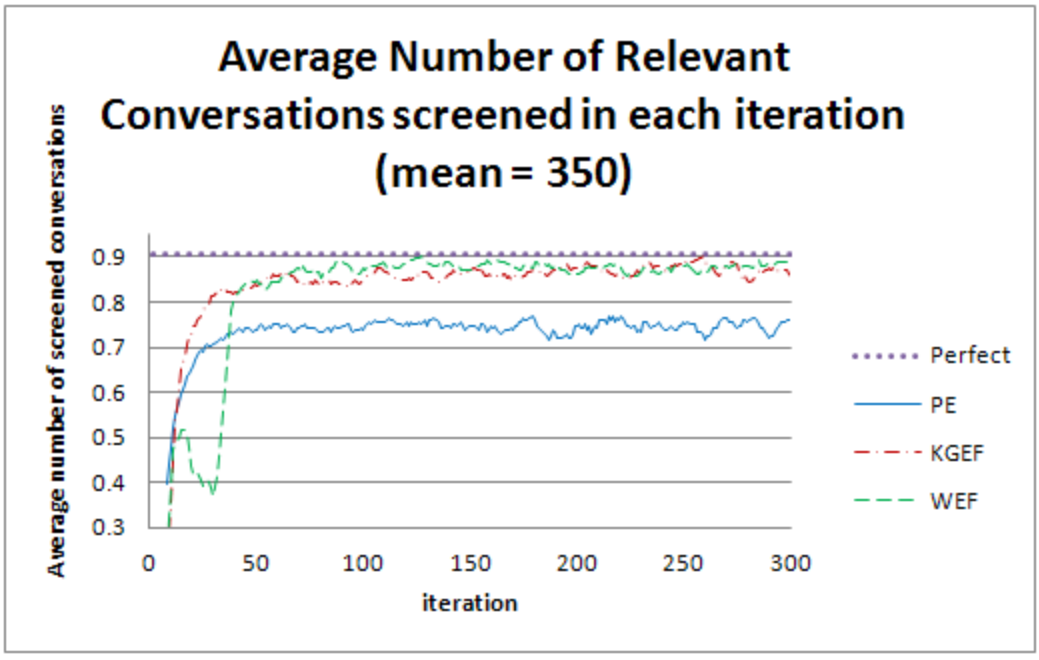


Figure 28. The value of $\overline{p}^{(k)}$ for the algorithms PE, KGEF and WEF with a mean of 350 conversations.

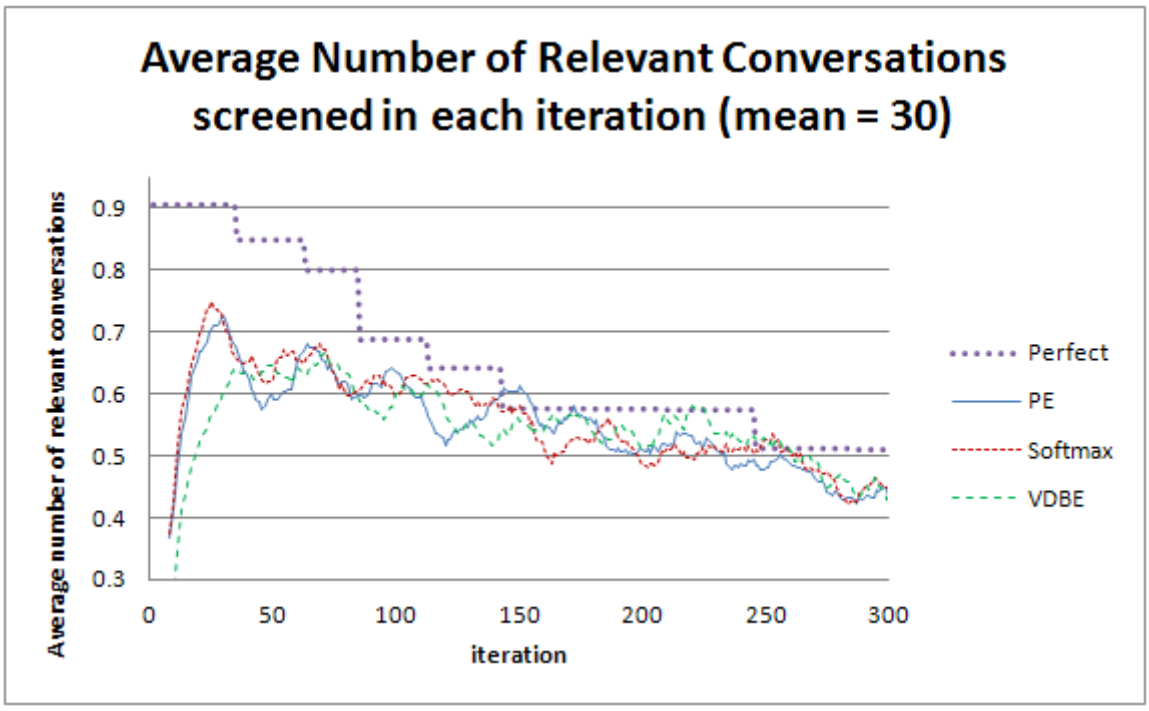


Figure 29. The value of $\overline{p}^{(k)}$ for the algorithms PE, Softmax and VDBE with a mean of 30 conversations.

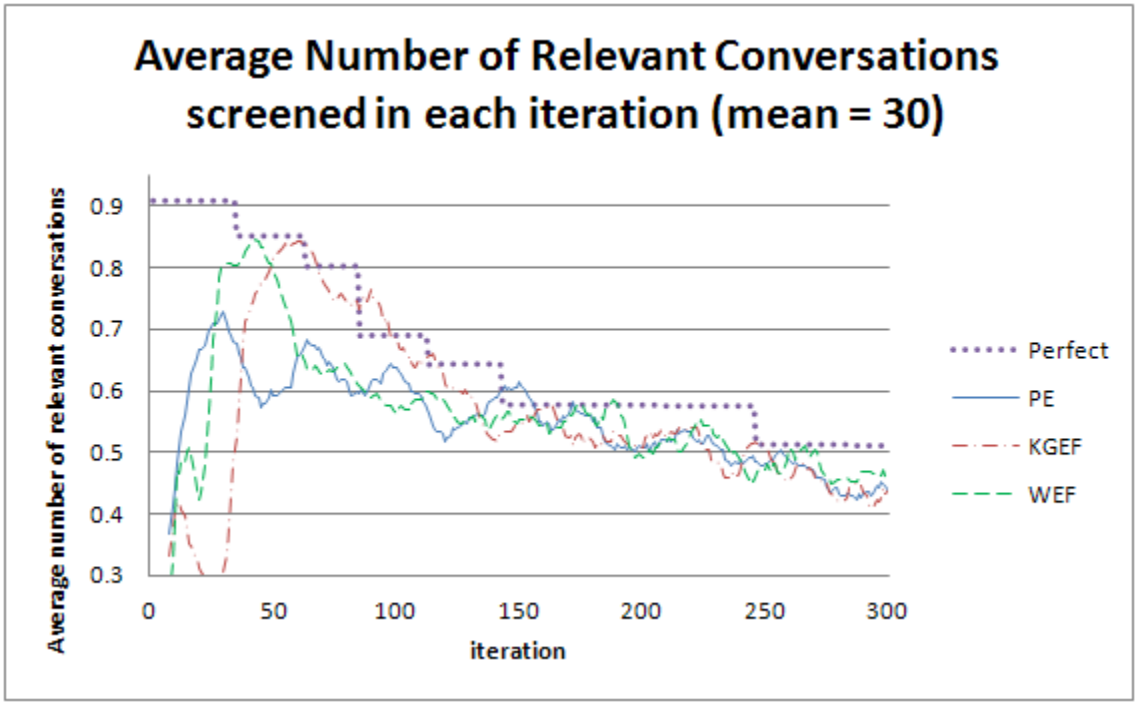


Figure 30. The value of $\overline{p}^{(k)}$ for the algorithms PE, KGEF and WEF with a mean of 30 conversations.

As in the previous section, we now compare the value of $\overline{p}^{(k)}$ for each algorithm. Figures 27 and 28 show the value of $\overline{p}^{(k)}$ when the mean number of conversations is 350. Those graphs are useful, as they show how fast and how accurate do the algorithms find the optimal edge. PE, for example, finds an edge very quickly but usually the edge it finds is a sub-optimal one. Both Softmax and VDBE have a similar accuracy, better than that of PE. Interestingly, they reach about the same level of accuracy as PE after 50 iterations, but then they keep gradually improving during the next 50 iterations as they get closer to the optimal edge. Both KGEF and WEF are very accurate (reach a $\overline{p}^{(k)}$ of almost 0.9). The KG algorithm gets gradually closer and closer to the optimal edge, while the WEF algorithm has a very low value of $\overline{p}^{(k)}$ during the exploration iterations, but then the value of $\overline{p}^{(k)}$ abruptly increases as the algorithm switches to exploitation.

When the mean number of conversations is 30, after 60 iterations all the algorithms perform pretty much the same. This is probably the reason why there is no significant change in the total number of relevant conversations and the standard deviation, as seen in Figures 25, 26. In addition, their value of $\overline{p}^{(k)}$ is close to that of the perfect algorithm.

2. Graph Topology

The graph in our analysis is created by adding dummy nodes to the terrorist network shown in Figure 7 of Chapter IV, and then adding edges randomly. Now, instead of the terrorist network in Figure 7 we use networks with different topologies. We maintain the same number of nodes in the network, and randomly add nodes and edges in the same way we did before (described in Chapter IV). We replace the terrorist network with a network composed of four separate cliques (each clique the size of four), and a network in which all the terrorists are forming a single line, i.e., the ith terrorist is connected to the $(i-1)$th and the $(i+1)$th terrorists. We refer to the graphs as *clique graph* and *line graph.* The resulting graphs, after adding nodes and randomly adding edges, are shown in Figure 31.

As in the previous section, we examine different possible values for the algorithms parameters, and choose the algorithms which produced the best results. The results are shown in the Table 26.

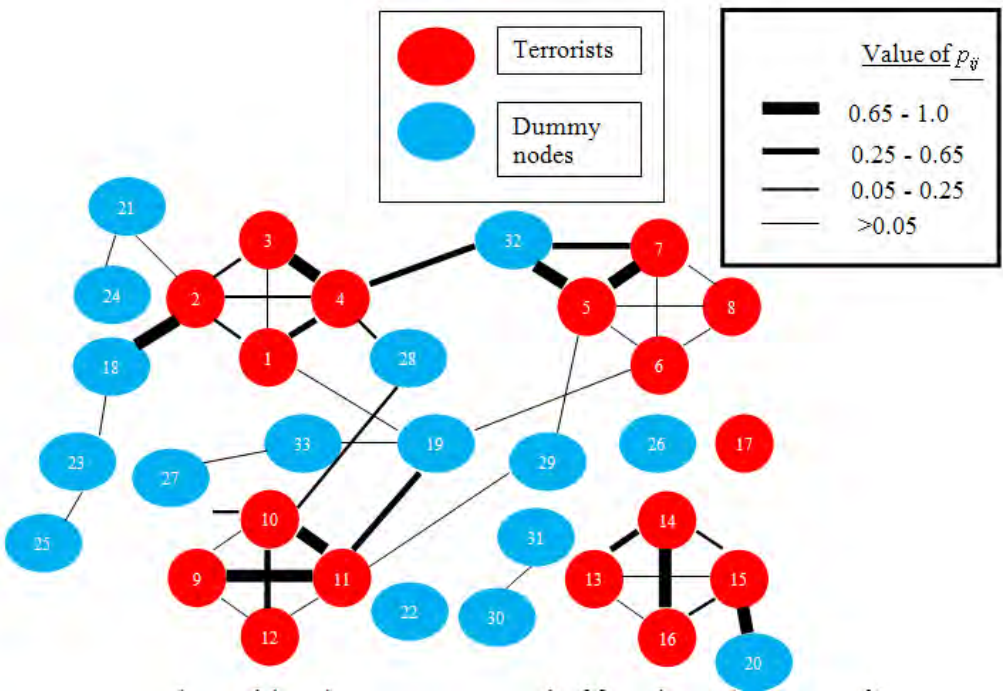
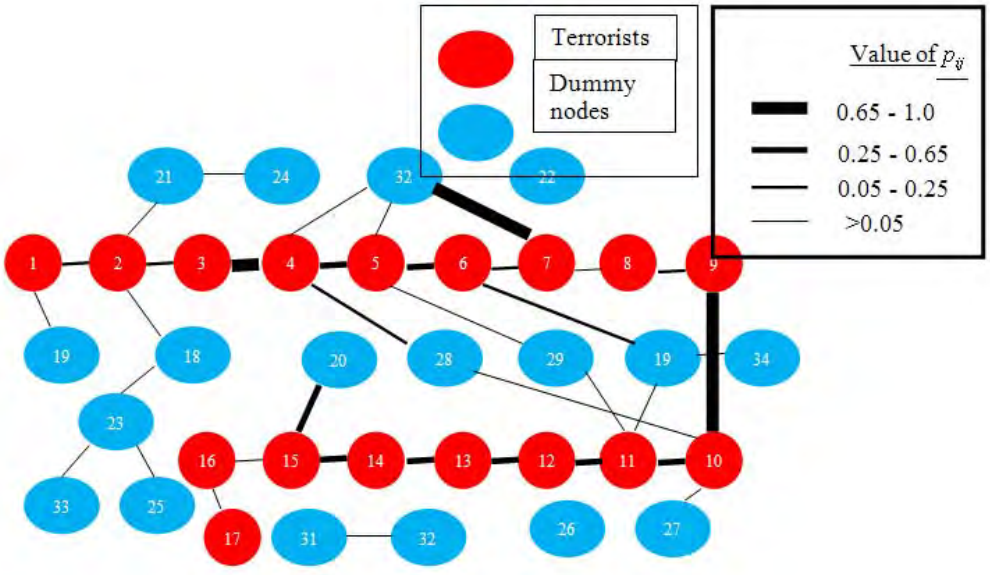


Figure 31. Up – the graph based on a terrorist network of four cliques (*cliques graph*). Down – the graph based on a terrorist network shaped as a single line (*line graph*).



The algorithm:	**The parameters:**	**Original Graph**	**Cliques Graph**	**Line Graph**
Pure Exploitation	None			
Softmax	Temperature	0.08	0.08	0.12
KG-VDBE	δ	0.1	0.15	0.05
	σ	0.4	0.3	0.3
WEF	# iterations of exploration	20	30	30
	# of samples from each edge (B)	3	3	2
KGEF	# iterations of exploration	40	30	50

Table 26. The algorithms parameters values given different graph topologies.

Table 26 shows that for the line graph, all the algorithms tend more towards exploration: the temperature parameter in Softmax is higher (0.12 instead of 0.08), the delta for the VDBE algorithm is significantly lower (0.05 instead of 0.1), and more exploration iterations are needed for both the WEF and KGEF algorithms (30 instead of 20 and 50 instead of 40).

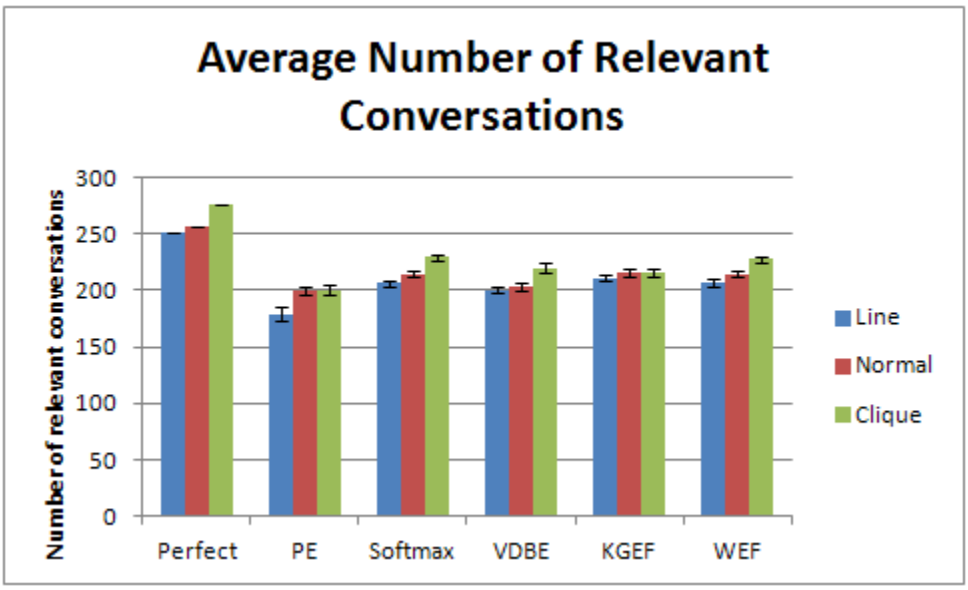


Figure 32. Comparison between the average number of relevant conversations screened by the algorithms, given different graph topologies.

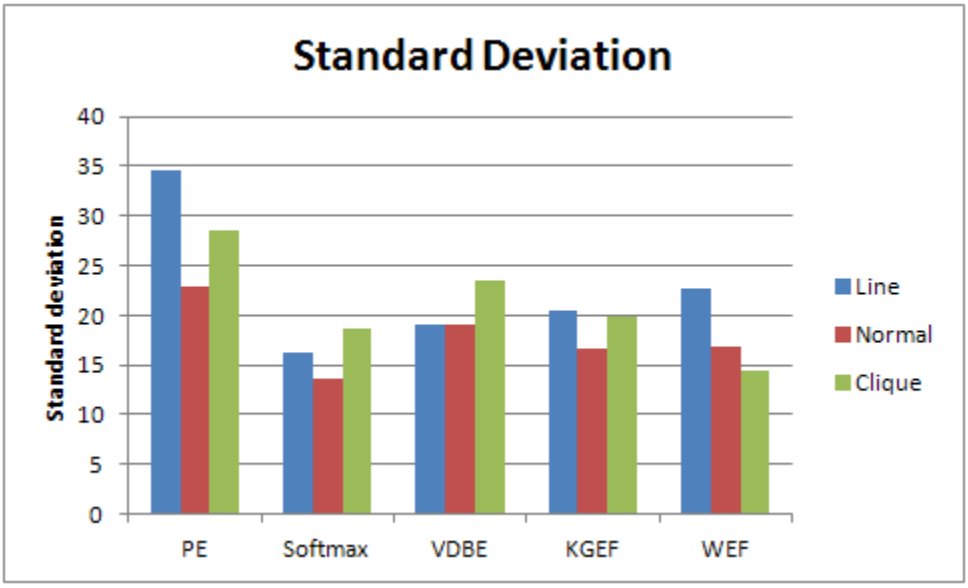


Figure 33. Comparison between the standard deviation of the algorithms, given different graph topologies.

The results are summarized in Figures 32 and 33. In both clique graph and line graph, PE has the worst performance (with 95% confidence). We start by examining the way the parameter $\overline{p}^{(k)}$ changes throughout the screening process. With the normal graph

(Figures 23, 24) the value of $\overline{p}^{(k)}$ for all algorithms during the last 50 iterations is almost the same, i.e., all algorithms perform the same as the PE algorithm. However, given the cliques graph (Figures 34, 35), all algorithms perform better than the PE algorithm throughout the entire screening process. The Softmax algorithm performs a little better than the VDBE algorithm at the beginning of the process, but after about 100 iterations their performance is pretty much the same. Interestingly, after the 50th iteration the WEF algorithm performs much better than the KGEF. In comparison, based on the normal graph after the 50th iteration the WEF and KGEF performed pretty much the same.

Given the line graph (Figures 35, 36), all algorithms perform better than the PE algorithm throughout the entire screening process. Generally, although the algorithms required a modification of their parameters, they all performed relatively well and showed a similar performance as to that shown in Figures 23 and 24 (the behavior given the normal graph).

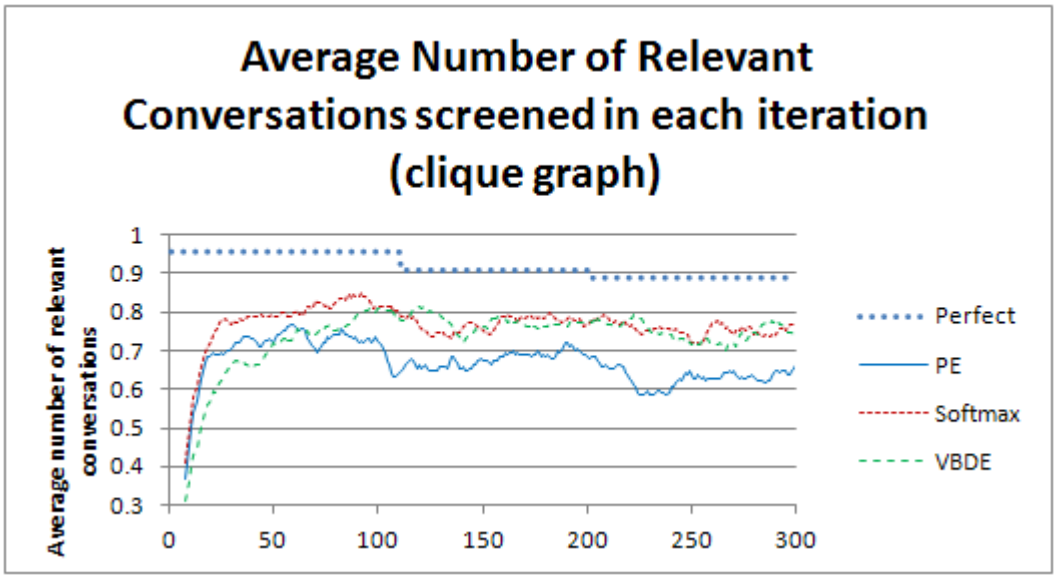


Figure 34. The value of $\overline{p}^{(k)}$ throughout the screening process for the algorithms PE, Softmax and VDBE, given a cliques graph

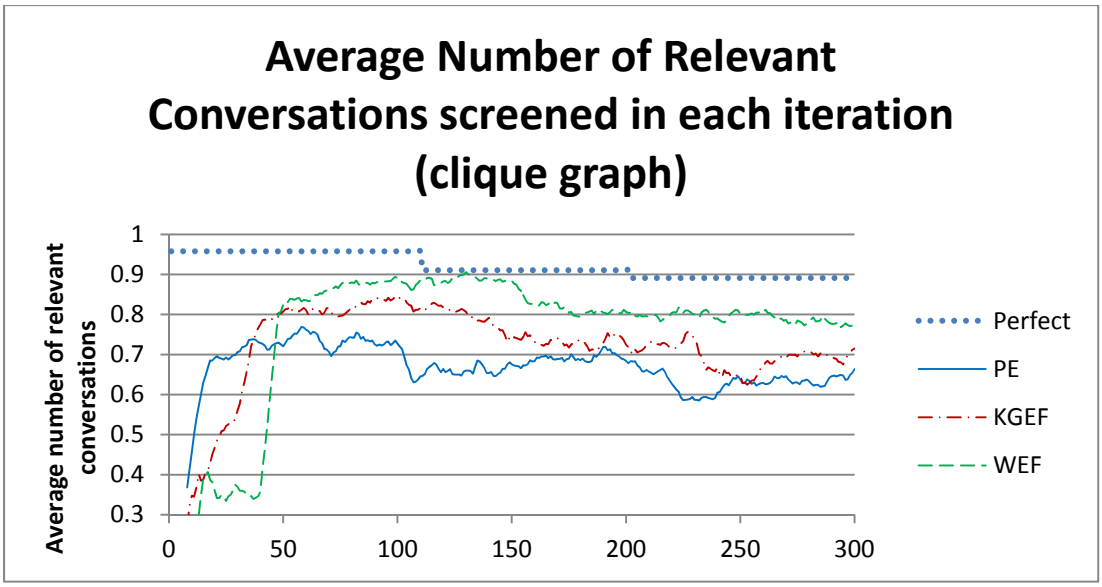


Figure 35. The value of $\overline{p}^{(k)}$ throughout the screening process for the algorithms PE, KGEF and WEF, given a cliques graph

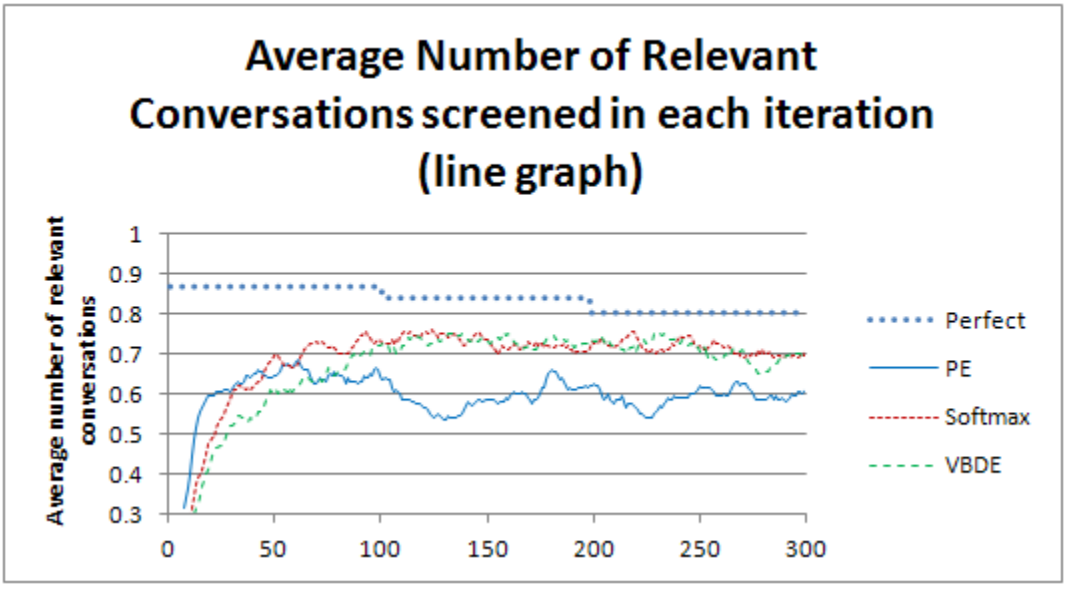


Figure 36. The value of $\overline{p}^{(k)}$ throughout the screening process for the algorithms PE, Softmax and VDBE, given a line graph

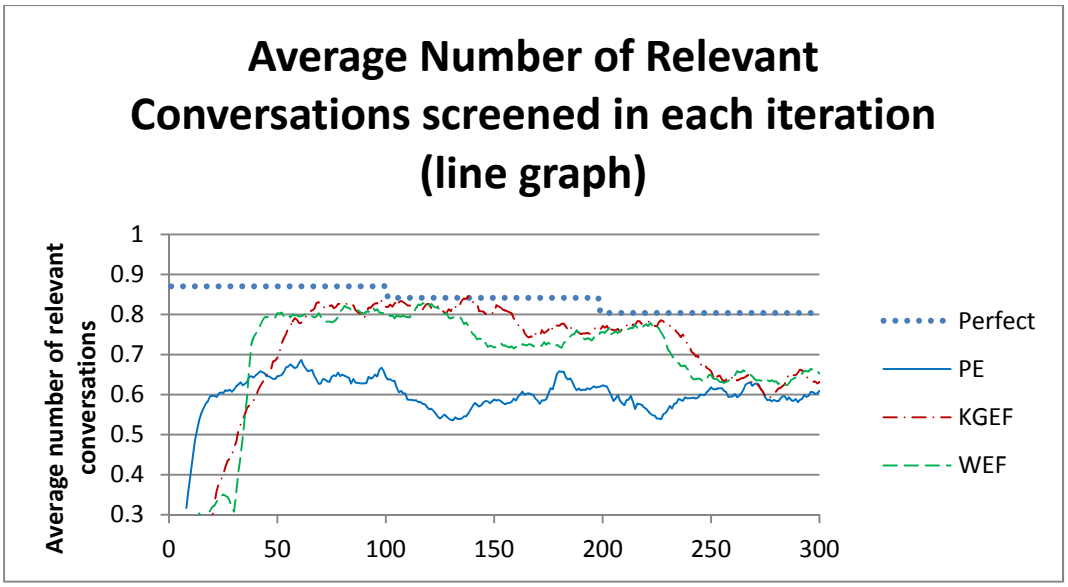


Figure 37. The value of $\overline{p}^{(k)}$ throughout the screening process for the algorithms PE, KGEF and WEF, given a line graph

D. ANALYSIS CONCLUSIONS

The analysis in Chapter V provides us with some insights regarding the performance of the different algorithms for the information selection problem proposed in this thesis.

1. Different Stages of the Screening Process

Based on Figures 23 and 24 we can divide the screening process into several main stages: The initial search after high value edges (i.e., edges with a high value of p_{ij}), and stages in which the algorithms focus on a single edge. Between focusing on different edges, there may be short periods in which the algorithm searches again for new edges.

For the PE, Softmax and VDBE algorithms, the length of the initial search stage may vary, and during that stage the average number of relevant conversations screened in each iteration (represented by the parameter $\overline{p}^{(k)}$ shown earlier in this chapter) gradually increases. For the KGEF and WEF algorithms, the length of the initial search period is fixed, and the value of $\overline{p}^{(k)}$ during this stage remains relatively low. The value of $\overline{p}^{(k)}$ for

the KGEF algorithm slightly increases during the initial search stage, unlike the value of $\overline{p}^{(k)}$ for the WEF algorithm which remains relatively constant.

There is a significant difference between the average value of p_{ij} for the first edge the algorithms focus on, and that of the edges the algorithm focus on later. Considering the first edge, the WEF and KGEF algorithms have a clear advantage over the other algorithms, as they tend to choose an edge whose value of p_{ij} is very close to the maximal possible. However, this advantage decreases as the number of iterations increases. Figures 23 and 24 show how the value of p_{ij} of the chosen edge gradually decreases between the first, second and third edges chosen, for all algorithms. For the third edge chosen, there is almost no difference between PE and the other algorithms.

The main conclusion is that the algorithms Softmax, VDBE, KGEF and WEF managed to identify one or two edges with a relatively high value of p_{ij}, but were usually unable to identify a third edge with this property. Changing the parameters of the VDBE, Softmax, KGEF and WEF algorithms to prefer exploration over exploitation should improve the ability of the algorithms to identify more edges with a high value of p_{ij}, but would increase the length of the initial search period and might therefore decrease the total number of screened conversations.

2. Performance of the Different Algorithms

a. The PE Algorithm

The PE algorithm showed the worse results compared to the other algorithms: a relatively low number of relevant conversations screened, and a significantly higher standard deviation (compared to the other algorithms). However, the PE algorithm still managed to achieve a $\overline{p}^{(k)}$ of 0.7–0.8 (as shown in Figure 23) after a relative short initial search period, which is rather impressive as there are only five edges with a value of p_{ij} above 0.65, and only two with a value larger than 0.8 out of almost fifty possible edges. In addition, the difference between the total screened number of conversations between the algorithms (as shown in Figure 21) was relatively small (less

than 10 relevant screened conversations). We therefore believe that the dependencies between the different edges have improved the performance of the PE algorithm. More generally, the correlation results in a preference towards exploitation. The reason is that due to the correlation, information regarding other alternatives can also be gained during the exploitation iterations and not only during the exploration iterations.

b. The Softmax Algorithm

Despite the fact that the Softmax algorithm is relatively simple, it has shown relatively nice results: It reached a comparatively large total number of relevant screened conversations in different scenarios (as shown in Figures 24, 34 and 39) and a very small standard deviation (as shown in Figure 22).

c. The VDBE Algorithm

The VDBE algorithm performed worse than expected. Tokic (Tokic, 2010) shows that the VDBE algorithm performs significantly better than Softmax. However, in our analysis the Softmax algorithm performed as well or better than the VDBE algorithm (as shown in Figures 21, 25 and 32). This can be explained by the correlation between the alternatives in our model. That correlation allows us to explore more efficiently, and the VDBE algorithm fails to take that into account. Another disadvantage of the algorithm is that it requires several input parameters, and it is relatively difficult to determine their optimal values (as explained in Chapter IV).

d. The KGEF Algorithm

The KGEF algorithm has several advantages. After a relatively small number of iterations (compared to the WEF algorithm, for example) it manages to identify and edge whose value of p_{ij} is close to the maximal possible. In addition, the KG policy requires no parameters, which is a clear advantage from a practical point of view. The main disadvantage of this algorithm is that it fails to identify more than one or two edges with a high value of p_{ij}.

e. The WEF Algorithm

As the KGEF algorithm, the WEF algorithm also manages to identify an edge with a very high value of p_{ij} after the initial search period. In different scenarios, it also showed a very small standard deviation (compared to the KGEF algorithm, for example). Since this algorithm is based on intuitive heuristics which might be employed by a real-life collector, the results of our analysis show that those intuitive heuristics might result in very good results.

3. Factors Affecting the Performance of the Algorithms

The mean number of relevant conversations clearly affects the performance of the algorithms, as seen in Figures 25 and 26. The difference between the algorithms performance was much more significant with a higher mean number of relevant conversations, and there was almost no difference between them when the mean number of conversations was 30 instead of 100. This result shows that exploration is only important when the collector is able to take advantage of the gained information.

The topology of the graph also affects the results, and might require changing some of the parameters–when we used the line graph, the algorithm parameters where modified to give preference to exploration. Further research is needed to draw more general conclusions regarding the way the topology affects the performance of the different algorithms. However, the algorithms behavior remains relatively consistent despite the changes in the graph topology. That outcome reassures us that our results would still be valid for different networks, aside from the case study we examined.

VI. CONCLUSION

In this chapter we summarize the results of our study, show possible extensions of the model, and propose several future research directions.

A. SUMMARY AND MAIN CONCLUSIONS

The collectors in the Processing and Exploitation stage (the third stage in the intelligence cycle) face the information selection problem: Which intelligence items to screen in order to maximize the expected amount of relevant information gained.

To handle this problem, we constructed a mathematical model of the intelligence items screening process, as manifested in the screening of intercepted conversations from a communication network (see Chapter II). This mathematical model is one of the main contributions of this research, mainly due to the lack of mathematical models for intelligence processes in the current (open) literature (see Chapter I). The model is fairly robust, and thus can be used to further analyze the Processing and Exploitation stage, beyond the specific problem presented in this research. Possible extensions of the model, and further research directions are presented later on in this chapter.

Based on this mathematical model, we examined several possible algorithms to handle the information selection problem. The algorithms are presented in Chapter III. To analyze the performance of these algorithms, we constructed a simulation of the screening process, as presented in Chapter IV. Using the simulation, we examined the performance of the algorithms given a specific scenario, based on the terrorist network responsible to the U.S. embassy bombing in Tanzania in 2007.

Our analysis, presented in Chapter V, provides some key insights on the information selection problem:

- Simple algorithms, both a simple greedy algorithm (PE) and Softmax, performed much better than anticipated. We assume that the dependencies among the alternatives are the main reason for that performance.

- The algorithms which consistently showed a good performance (even after changing some of the simulation parameters) are the WEF, an intuitive heuristic which can be easily employed in practice by a collector, and KGEF, an algorithm based on the Knowledge-Gradient policy.
- The mean number of conversations in each edge is a significant factor that affects the performance of the algorithms. When the mean number of conversations is small, there is no significant difference between the performance of the simple greedy algorithm and that of the other, more sophisticated and complex, algorithms.

B. POSSIBLE EXTENSIONS OF THE MODEL

We now propose several possible ways in which the assumptions of the model can be relaxed. The order in which the extensions are shown is in accordance with their complexity: We start with extensions which only require few changes in the model, and move on to more complicated extensions.

1. Prior Knowledge of the Collector

Our model assumes a prior Uniform distribution over the different possible relevance values. A general distribution, representing a prior knowledge of the relevance values, can be easily used instead.

2. Identified and Unidentified Relevance Conversations

In Chapter II, we list several assumptions regarding identifying the relevance values of the nodes:

- The relevance values of the nodes are either identified or unidentified. An unidentified node can only become identified if the collector listens to a conversation in which it participates.
- The probability of identifying the relevance value is fixed (c), and is independent of the relevance value itself (d_i).

- If both nodes in the conversation are unidentified, then the probabilities of identifying the relevance value of the nodes are independent of each other.

The following relaxations of the assumptions require only minor changes in the model:

- A node might be identified by listening to conversations which do not include that node. This relaxation represents the possibility that in a conversation a person might provide information about another person. We can therefore assign a probability c' that the relevance value of a certain node i is identified in the conversation. We might adjust this assumption so that only neighbors of the nodes participating in a conversation can be identified.
- The probability to identify node i might depend on its unknown relevance value d_i, i.e., instead of a constant c we can use the function $c(d_i)$.

If both nodes in a conversation are unidentified, the probabilities to identify them might not be independent. All those changes only affect the way the simulation determines whether a node is identified or not. It only affects algorithms which take the possibility of identifying a node into account: amongst the algorithms listed in Chapter III, only the KGEF algorithm would be affected.

A more complex change would be a relaxation of the assumption that a node is either identified or unidentified. In our model, when the node is unidentified its relevance value is estimated using a certain probability distribution. However, based on information from the content of the conversations (e.g. the profession of the person represented by the node), the node might be identified and then the collector knows its relevance value (d_i). Instead, we might argue that the exact relevance value always remains unknown. Information gathered from the content of the conversations will then only change the probability distribution over the possible relevance values (i.e., the distribution of the

random variable D_i). In order to take that change into account, one needs to define how exactly relevant information might affect the probability distribution of the random variable D_i.

3. Conversations with Different Values

We assume that a conversation might be either relevant or irrelevant. Therefore, all the relevant conversations in our model have the same operational value. However, some relevant conversations might be more valuable than others. In our model terminology, the random variables $S_{ij}^{(k)}$ (which represent the relevance of a conversation screened between nodes i and j in the k th iteration) might have several possible values, not just zero and one.

To allow different values of conversations in our model, we need to define the density functions of the random variable $S_{ij}^{(k)}$, and the way it depends on the relevance values of the nodes. In our model, we made sure that the parameters of the random variables $S_{ij}^{(k)}$ are drawn from a conjugate prior distribution, in our case a Beta distribution (as explained in Chapter II). Generally, the distribution from which the parameters are drawn does not have to be a conjugate prior. However, having a conjugate prior distribution simplifies the model, as keeping track of the distribution during the updating process becomes much easier. In our model, for example, without a conjugate prior distribution we would have needed to keep track of the entire distribution of the different P_{ij}, a distribution which constantly changes throughout the updating process. However, we only need to keep track of the discrete functions $\alpha(d_i, d_j), \beta(d_i, d_j)$, and this suffices to determine the distributions of the P_{ij}.

Fink (Fink, 1997) suggests several possible conjugate functions. Suppose, for example, that instead of a Bernoulli random variable $S_{ij}^{(k)}$ we want to use a Binomial random variables with the a known set of integers $\{0,1,\ldots,m\}$ as possible values, and an

unknown parameter p_{ij}. Then, according to Fink, a Beta distribution (as used in our current model) is a possible conjugate distribution from which the values of p_{ij} can be drawn.

This modification would affect the model and the updating process, but all the different algorithms mentioned in Chapter III can still be used with no significant changes.

4. Time Dependent Conversation Values

We assume that: 1) the collector faces a strict time constraint, i.e., he can screen no more than T conversations; 2) the value of the information gained is independent on the time in which it is gained. However, there are scenarios in which the earlier the information is gained the more valuable it is. For example, if the information is needed to support some operational activity. Many exploration-exploitation models (e.g. Tokic, 2010, Frazier et al., 2010 and Gittins et al., 2011) encompass this by multiplying the reward of an alternative (in our case, the value of the conversation) with some discount factor γ^k, where γ is a constant between zero and one, and k is the number of iterations. The value of a conversation in the k th iteration is therefore: $\gamma^k S_{ij}^{(k)}$.

The updating process can be modified relatively easily. If the value of a conversation in the k th round is v, then $S_{ij}^{(k)} = \frac{v}{\gamma^k}$, and the updating process can be performed accordingly. However, some of the algorithms in Chapter III need to be modified. The algorithms Pure Exploitation, Softmax, VDBE and the Wide Search policy would remain the same. The Exploration First algorithm would remain the same, but the optimal number of exploration rounds might vary, depends on the discount factor γ. The Knowledge Gradient policy is needed to change, where the "future value" described in chapter III is replaced with the maximal $\mathrm{E}[P_{ij}]$ (as in Frazier et al., (Frazier et al., 2010)).

In addition, if we remove the strict time constraint of only T conversations, it is possible to include some form of the Gittins indices (Gittins et al., 2011) in the algorithms. The Gittins indices policy requires an infinite time horizon, a constant

discount factor γ and independence between the alternatives (Gittins et al., 2011). Although generally the alternatives are dependent, after the relevance values of the nodes i and j are identified, the conversations from the edge (i, j) are independent of the other nodes and edges. Therefore, if exploitation is only performed on conversations between nodes whose relevance value are identified, then the assumptions for using the Gittins indices hold.

5. Decreasing Value of Conversations from the Same Edge

A main reason to screen information from multiple sources is that the information from the same source might repeat itself. One way to model that is to multiply the value of a conversation between i and j with a discount factor $\lambda^{k_{ij}}$, where λ is a discount factor and k_{ij} is the number of screened conversations between nodes i and j.

If the value of a conversation in the k th round is v, then $S_{ij}^{(k)} = \frac{v}{\lambda^{k_{ij}}}$, and the updating process can be performed accordingly. In each one of the algorithms, the conversations in each iteration should be chosen based on $\mathrm{E}[S_{ij}^{(k)}]$ instead of $\mathrm{E}[P_{ij}]$.

6. Using the Model for a Large Scale Network

We only used a relatively small scale network. The reason is that performing the variable elimination method (explained in Chapter II) for a large graph requires a very long time. In order to use the model for a large-scale network, we need an inference method that would replace the variable elimination method. Kohler and Friedman show approximate inference methods that can be applied instead of the variable elimination method (Kohler and Friedman, 2010). Those methods require significant changes in the updating process, as they do not use factors as representations of the joint distributions. However, no changes are required for the algorithms in Chapter III.

C. FUTURE RESEARCH

1. Broad Experiments

Our analysis methodology in Chapter IV and Chapter V allows us to recieve insights regarding the information selection problem and the different algorithms examined. However, our analysis does not enable us to provide general answers to general questions. For example, we cannot thoroughly answer the question how the network topology affects the performance of the algorithms, although we have some insights regarding that questions (as mentioned in section A). In order to draw such general conclusions, our model can be used as a basis for a more broad experiment than the one shown in Chapter V (for example, examine many randomly generated graph topologies instead of the three examples shown in Chapter V). Therefore, a future research direction is to use our model in order to draw general conclusions about the Processing stage.

2. Advanced Algorithms

Another future direction would be to create better algorithms to solve the problem. Our problem concentrated on examining the performance of known algorithms and intuitive heuristics. A further research might be more focused on improving the existing algorithms, or developing new algorithms to handle the information selection problem. Our analysis on Chapter V can be provide a better understanding of the existing algorithms and heuristics, and therefore be a starting point for the development of advanced algorithms.

3. Real-World Data

The values of the simulation parameters we used (shown in Chapter IV) is not based on real-world data. Real-world data would allow determining more realistic values for the different parameters. The modifications in the previous section might also be used to turn the model into more realistic.

4. Further Modifications of the Model

We now propose several extensions of the model. Unlike the extensions mentioned in the previous section, those extensions require a reformulation of the problem.

a. Screening Conversations with Different Lengths

One of our assumptions is that screening every conversation requires the same amount of time. Therefore, the time constraint T is an integer describing the number of conversations the collector can screen. However, different conversations might require different amount of time to screen (depends, for example, on the length of the conversations). In addition, the probability that a conversation is relevant might depend on the time needed to screen it.

Currently, choosing a conversation mostly depends on the expected probability that the conversation is relevant ($\mathrm{E}[P_{ij}]$). With different screening times, the length of the conversation is another criterion that needs to be taken into account. That problem resembles a dynamic and stochastic knapsack problem (Kleywegt et al., 1998). Therefore, significant changes in the algorithms and further research are required to solve this problem.

b. Including Errors in the Model

Costica (Costica, 2010) analyzes errors in determining whether a conversation is relevant or irrelevant. However, his analysis does not take into account the information selection problem we presented. In our analysis we ignore the possibility of errors in the screening process. Such errors might include:

- Errors in determining that a conversation is relevant or irrelevant (either false-positive or false-negative errors).
- Errors in identifying the relevance values of the nodes.
- Errors in the prior joint distribution: the parameters d_i, p_{ij} are drawn from a different prior joint distribution than the distribution of the collector.

To include the possibility of errors in the model significant changes in the model and in the algorithms may be required. That analysis is beyond the scope of this research, and further research is needed.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Bellman, R. (1957). *Dynamic programming.* Princeton: Princeton University Press.

Berry, D. A., & Fristedt, B. (1985). *Bandit Problems*. Michigan: Chapman and Hall.

Boutilier, C. (2002), "A POMDP formulation of preference elicitation problems," in *Proceedings of the National Conference on Artificial Intelligence,* Edmonton, AB 239–246.

Bron, C., & Kerbosch, J. (1973), "Algorithm 457: Finding all cliques of an undirected graph," *Commun. ACM (ACM)* 16(9): 575–577.

Cassandra, A., Kaelbling, L., & Littman, M. (1994). "Acting optimally in partially observable stochastic domains," in *Proceedings of the Twelfth National Conference on Artificial Intelligence,* (AAAI) Seattle, WA.

Central Intelligence Agency. (1999). *Consumer's guide to Intelligence*.

Cohen, J., McClure, S., & Yu, A. (2007). "Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration," *Philosophical Transactions*, 362, 933–942.

Costica, Y. (2010). *Optimizing classification in intelligence processing* (master's thesis). Operations Research department, Naval Postgraduate School, Monterey, CA.

Daw, N. D., O'Doherty, J. P., Seymour, B., Dayan, P., & Dolan, R. J. (2006). "Cortical substrates for exploratory decisions in humans," *Nature* 441, 876–879.

Desimone, R., & Charles, D. (2002). "Towards an ontology for intelligence analysis and collection management," in *Proc 2nd International Conference on Knowledge Systems for Coalition Operations (KSCO 2002)*, 2002, 26–32.

Devore, J. L. (2009). *Probability and statistics for engineering and the sciences* (7th ed.). Boston: Duxbury Press.

Erdos, P., & Renyi, A. (1959). "On Random Graphs," *Publicationes Mathematicae* 6, 290–297.

Fink, D. (1997). "*A compendium of conjugate priors*," Technical Report, 1997.

Frazier, P., Powell, W., & Dayanik, S. (2009). "The Knowledge-Gradient Policy for Correlated Normal Beliefs," *INFORMS journal on Computing*, *21*(4), Fall 2009, 591–613.

Fu, M. C., Hu, J. Q., Chen, C. H., & Xiong, X. (2007). "Simulation allocation for determining the best design in the presence of correlated sampling," *INFORMS Journal on Computing 19*(1), Winter 2007, 101–111.

George, E., Makov, U., & Smith, F. (1993). "Conjugate Likelihood Distributions," Scandinavian Journal of Statistics, *20*(2), 1993, 147–156.

Gill, P. (2007). "Sorting the wood from the trees," in: Johnson, L.K., *Strategic Intelligence: understanding the hidden side of government*, Praeger, 2007.

Gittins, J., Glazebrook, K., & Weber, R. (2011). *Multi-Armed Bandit Allocation Indices*, Wiley.

Hammersley, J. M., & Clifford, P. (1971). "*Markov Fields on Finite Graphs and Lattices*." Unpublished.

Hedley, J. H. (2007). "Analysis for strategic intelligenc," in Johnson, L.K., *Strategic Intelligence: understanding the hidden side of government*. Santa Barbara, CA: Praeger.

Hulnick, A. S. (2006). "What's wrong with the Intelligence Cycle?" *Intelligence and National Security*, 21(6), 959–979.

Johnson, L. K. (2007). *Handbook of intelligence studies*. New York: Routledge.

Kaplan, E. H. (2011), "Operations research and intelligence analysis," in: Fischhoff B. and Chauvin C., *Intelligence analysis: behavioral and social scientific foundations*, NAP, Washington, D.C. 2011.

Kleywegt, A. J., & Papastavrou J.D. (1998). "The dynamic and stochastic knapsack problem," *Operations Research*, 46, 17–35.

Koller, D., & Friedman, N. (2010). *Probabilistic Graphical Models*, MIT press.

Langville, A., & Meyer, C. (2005). "A survey of eigenvector methods for web information retrieval," *SIAM review*, *47(*1), 135–161.

Lindelauf, R., Borm, P., & Hamers, H. (2008). "On Heterogeneous Covert Networks," *CentER Discussion Paper Series*, 46, April.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

McPherson, M., Smith-Lovin, L., & Cook, J. (2001). "Birds of a Feather: Homophility in social networks," *Annu. Rev. Social*, *27*, 415–44.

Preece, A., Pizzocaro, D., Borowiecki, K., de Mel, G., Gomez, M., Vasconcelos, W., Bar-Noy, A., Johnson, M. P., La Porta, T., Rowaihy, H., Pearson, & G., Pham, T. (2008). "Reasoning and resource allocation for sensor-mission assignment in a coalition context," in *sub. To Milcom 08*, July 2008.

Richelson, J. T. (2007). "The technical collection of intelligence," in Johnson, L.K., *Handbook of intelligence studies*, 105–117. New York: Routledge.

Skroch, E. M. (2004). *How to optimally interdict a belligerent project to develop a nuclear weapon* (master's thesis). Operations Research department, Naval Postgraduate School, Monterey, CA.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction.* Cambridge, MA: MIT Press.

Thrun, S. B. (1992). "The role of exploration in learning control" in *Handbook of intelligent control: Neural, fuzzy, and adaptive approaches* (eds. D. A. White & D. A. Sofge), 527–559. Florence, KY: Van Nostrand Reinhold.

Tokic, M. (2010). "Adaptive epsilon-Greedy Exploration in Reinforcement Learning Based on Value Difference." In *Proceedings of KI 2010*. 203–210.

Whaley, B. (1974). *Codeword Barbarossa.* Cambridge, MA: MIT Press.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor Moshe Kress
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California

4. Professor Nedialko Dimitrov
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California

5. Professor Michael Atkinson
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California

6. Yeo Tat Soon
   Director, Temasek Defense Systems Institute (TDSI)
   National University of Singapore
   Singapore

7. Ms Tan Lai Poh
   Temasek Defense Systems Institute (TDSI)
   National University of Singapore
   Singapore