AFFINE ARITHMETIC-TYPE TECHNIQUES FOR HANDLING UNCERTAINTY IN EXPERT SYSTEMS

SANJEEV CHOPRA

Department of Computer Science

APPROVED:

Vladik Kreinovich, Chair, Ph.D.

Martine Ceberio, Ph.D.

G. Randy Keller, Ph.D.

Charles H. Ambler, Ph.D.
Dean of the Graduate School

to my

MOTHER, FATHER, Sister and Fiancée

with love

AFFINE ARITHMETIC-TYPE TECHNIQUES FOR HANDLING UNCERTAINTY IN EXPERT SYSTEMS

by

SANJEEV CHOPRA

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Computer Science

THE UNIVERSITY OF TEXAS AT EL PASO

July 2005

Abstract

Expert knowledge consists of statements S_j:facts and rules. The expert's degree of confidence in each statement S_j can be described as a (subjective) probability. For example, if we are interested in oil, we should look at seismic data (confidence 90%); a bank A trusts a client B, so if we trust A, we should trust B too (confidence 99%). If a query Q is deducible from facts and rules, what is our confidence $p(Q)$ in Q?

We can describe Q as a propositional formula F in terms of S_j; computing $p(Q)$ exactly is NP-hard, so heuristics are needed.

Traditionally, expert systems use technique similar to straightforward interval computations: we parse F and replace each computation step with corresponding probability operation. The problem with this approach is that at each step, we ignore the dependence between the intermediate results F_j; hence intervals are too wide. For example, the estimate for $P(A \vee \neg A)$ is not 1.

In this thesis, we propose a new solution to this problem; similarly to affine arithmetic, besides $P(F_j)$, we also compute $P(F_j \,\&\, F_i)$ (or $P(F_{j_1} \,\&\, \ldots \,\&\, F_{j_k})$), and on each step, use all combinations of l such probabilities to get new estimates. As a result, for the above stated e.g., $P(A \vee \neg A)$ is estimated as 1.

Table of Contents

Chapter 1

Introduction

1.1 Why Do We Need Expert Systems

In many application areas (e.g., in medicine, geology, economy, engineering), we often need to make decisions in the situation when we do not have the exact knowledge of the situation, and therefore, we cannot even formulate (not to say solve) the decision problems in precise mathematical terms. In some cases, we can formulate the problem precisely, but this formulation leads to a complicated mathematical optimization problem of the type that we cannot yet solve (e.g., control problems are in general computationally intractable, and therefore, we need experts to solve them [26, 40, 49]).

In all these cases, we need *human expertise* to make decisions. There often exist experts who make reasonably good decisions: expert doctors successfully cure diseases; expert geologists find oil; expert astronauts know how to dock and land the Space Shuttle; expert operators know how to operate a chemical plant, etc.

Usually, among the experts in the field, there are a few *top* experts whose decisions turn out to be the most efficient. Since there are only a few top experts, they cannot solve all the emerging problems. So, for all other problems, we have to rely on the expertise of those experts who may not possess all the knowledge of the top ones. It is therefore desirable to create an automated system that would incorporate the knowledge of top experts and use it to respond to the emerging problems in a similar way as the top experts would. Such a system would help other experts and lay people in making decisions. These systems are called *expert systems*.

In control, there is additional reason for introducing expert systems: it is often desirable to avoid using an expert altogether. Indeed, experts are expensive to use, error-prone, and in some dangerous environments, it is desirable to make completely automatic control systems: e.g., we want to control robots who travel into the volcanos or go to other planets.

1.2 Expert Knowledge

Expert systems contain expert knowledge. Expert knowledge usually consists of facts and rules. The main objective is, given a query Q, to check whether Q follows from the expert knowledge. For example, in the knowledge base

$$S_1 : a \leftarrow b.$$

$$S_2 : b \leftarrow .$$

$$S_3 : a \leftarrow c.$$

$$S_4 : c \leftarrow .$$

S_1 and S_3 are rules, and S_2 and S_4 are facts. If we ask a query $Q \stackrel{\text{def}}{=}$ "a?", then the answer is "yes": e.g., Q follows from S_1 and S_2. Prolog-type inference engines are tools that provide such inference; see, e.g., [37, 43].

1.3 Degrees of Belief (Subjective Probabilities) and Why They Are Necessary to Describe Expert Systems

The goal of an expert system is to simulate the experts' way of making decisions. For that purpose, an expert system includes a knowledge base that contains the knowledge of the experts. Some statements from the knowledge base are believed to be absolutely correct. However, in their decisions, experts also use other statements, about which they are not 100% sure that they are correct, and/or which are not formulated in exact terms. For example, their knowledge can contain phrases like "If a patient has a fever, a sore throat, and a headache, then most probably, he has a common cold". To describe the expert knowledge adequately, we must therefore store in the knowledge base not only the statements themselves, but also the indication of the extent to which the experts believe in these statements.

An intermediate degree of belief means that in addition to "true" and "false", we must label some statements by some intermediate labels, like "most probably true", or "probably false", etc. Inside the computer, "true" corresponds to 1, and "false" to 0. Therefore, it is natural to use numbers from the interval [0,1] to describe the intermediate degrees of belief. So, to describe the expert's degree of belief in a statement S, we must find an appropriate number $d(A)$ from the interval [0,1] (some expert systems use a different interval, e.g., $[-1, 1]$ [15, 39, 47, 48]).

There are several ways to assign $d(S)$ (see, e.g.,[21, 28, 41, 53, 57]):

- We can ask an expert to estimate his degree of belief on a scale from 0 to 10. If he, e.g., selects 8, then we take $d(S) = 8/10$.

  This is the simplest and the most straightforward method of eliciting the degree of belief from the experts. This method works well in many cases, but for some experts (e.g., for most mathematicians), it is difficult to think this way: they either know that a statement is true, or that it is false, or that they do not know. For such experts, different methods are necessary.

- If we have sufficiently many experts, and they are all definite in their beliefs, then we can poll the experts. For example, if 8 out of 10 experts believe that S is true, we assign $d(S) = 8/10$.

  Polling is fine if we have many experts with reasonably definite ideas. However, in many case, we have just a few experts, who are not certain about the statement

S, and who do now feel comfortable assigning a number to describe their degree of belief. Our ultimate goal is to make an expert system that makes decisions as well as these experts. So, these experts' degrees of belief are indirectly revealed by the decisions they make. Hence, to elicit the degrees of belief, we can simulate different situations, ask experts what decisions they would have made in these situations, and elicit the degree of belief from these decisions.

- One way of doing it is to ask an expert to choose between an alternative $\mathcal{S}$ in which he will receive \$100 if S is true and 0 otherwise, and a lottery $L(p)$ in which he gets \$100 with some given probability p. If for some p, these two alternatives are equivalent to an expert, we can say that his degree of belief in S is equal to this probability p. To determine p, we can use the following bisection procedure (see, e.g., [31]):

  - Initially, we know nothing about $d(S)$. We can express this by saying that the interval $[d^-, d^+]$ of possible values of $d(S)$ coincides with $[0, 1]$.
  - If we know that $d(S) \in [d^-, d^+]$, then we compare the alternative $\mathcal{S}$ and a lottery $L(p)$ with $p = (d^- + d^+)/2$. If $\mathcal{S}$ is preferable to $L(p)$, this means that for this expert, the degree of belief in S is larger than p. So, we can take $[p, d^+]$ as a new interval of possible values of degrees of belief. If $L(p)$ is preferable, this means that $d(S) < p$, and hence, we can take $[d^-, p]$ as a new interval of possible values of $d(S)$. In both cases, we get a new interval that is half the size of the original one. If we start with an interval $[0, 1]$ of size 1, and repeat this procedure k times, we get an interval of size 2^{-k}, i.e., we have determined $d(S)$ with an accuracy 2^{-k}.

1.4 How to Handle the Uncertainty in Expert Systems?

Since the statements S_i are true with certain probabilities $p(S_i)$, a natural question is: if a query Q is deducible from facts and rules, what is our confidence $p(Q)$ in Q [12, 13, 34]? For example, to find oil, we must look for certain subterranean structures (confidence 80%); to find these structures, we must analyze gravity data (confidence 90%). What is our confidence that to find oil, we must analyze gravity data?

Let us describe this problem in detail. We can describe deducibility of Q as a propositional formula F in terms of S_j. For example, for the above knowledge base, for Q to be true, either S_1 and S_2 must be true, or S_3 and S_4 must be true. In this case, $F = (S_1 \,\&\, S_2) \vee (S_3 \,\&\, S_4)$. The general algorithm for describing such a propositional formula is given in [35].

As a result, we arrive at the following problem:

- we have a propositional combination F of known statements S_j;
- we know the probabilities $p(S_j)$ of different statements;
- we must determine the probability $p(F)$.

Since the events S_j may be statistically dependent, we may get different values for $p(F)$ depending on whether the values are independent or, say, positively correlated.

1.5 Traditional Expert System Approach: AND and OR Operations With Degrees of Belief

Suppose that an expert system contains statements A and B, and we have elicited the degrees of belief $d(A)$ and $d(B)$ from the experts. Suppose now that a user wants to know the degree of belief in a composite statement $A \,\&\, B$.

In principle, knowing only the two numbers $d(A)$ and $d(B)$ is not sufficient to describe the expert's degree of belief in $A \,\&\, B$: e.g., if $d(A) = d(B)$, it could be that an expert perceives A and B as equivalent, in which case $d(A \,\&\, B) = d(A)$, or as independent, in which case the possibility of A *and* B being true is smaller than the possibility that one of them is true: $d(A \,\&\, B) < d(A)$. So, the ideal situation would be to elicit from an expert not only the degree of belief in the basic statements from the knowledge base, but also in all possible logical combinations of these statements. However, this is practically impossible: if we have N statements $S_1, ..., S_N$ in the knowledge base, then for each of $2^N - 1$ non-empty subsets $\{S_{i_1}, ..., S_{i_k}\}$ of the knowledge base, we need to elicit the degree of belief in the corresponding AND-statement $S_{i_1} \,\&... \&\, S_{i_k}$. For a realistic expert system, N is in hundreds, so asking an expert 2^N questions is impossible.

Therefore, although in some cases, we will be able to have the degree of belief $d(A \,\&\, B)$ stored in the knowledge base, in general, we often have to deal with a following situation: we know the degrees of belief $d(A)$ and $d(B)$ in statements A and B, we know nothing else about A and B, and we are interested in the (estimated) degree of belief of the composite statement $A \,\&\, B$. Since the only information available consists of the values $d(A)$ and $d(B)$, we must compute $d(A \,\&\, B)$ based on these values. We must be able to do that for arbitrary values $d(A)$ and $d(B)$. Therefore, we need a *function* that transforms the values $d(A)$ and $d(B)$ into an estimate for $d(A \,\&\, B)$. Such a function is called an *AND-operation*. If an AND-operation $f_\& : [0,1] \times [0,1] \to [0,1]$ is fixed, then we take $f_\&(d(A), d(B))$ as an estimate for $d(A \,\&\, B)$.

Similarly, to estimate the degree of belief in $A \vee B$, we need an *OR-operation* $f_\vee : [0,1] \times [0,1] \to [0,1]$.

These operations must satisfy some *natural conditions*. For example, for an expert, $A \,\&\, B$ and $B \,\&\, A$ mean the same. Therefore, the estimates $f_\&(d(A), d(B))$ and $f_\&(d(B), d(A))$ for these two statements should coincide. To achieve that, we must require that $f_\&(a, b) = f_\&(b, a)$ for all a and b; in other words, the operation $f_\&$ must be *commutative*. Similarly, from the fact that $A \,\&\, (B \,\&\, C)$ and $(A \,\&\, B) \,\&\, C$ mean the same, we can deduce the requirement that $f_{\&}$ must be *associative.* If A is absolutely false ($d(A) = 0$), then $A \,\&\, B$ is also absolutely false, i.e., $f_{\&}(a, 0) = 0$ for all a. If A is absolutely true $d(A) = 1$, then $A \,\&\, B$ is true iff B is true, so, the degree of belief in $A \&, B$ must coincide with the degree of belief in B: $d(1, b) = b$ for all b. If our belief in A or B increases, then the degree of belief in $A \,\&\, B$ must also increase, so $f_{\&}$ must be *monotonic.* If the degree of belief in A changes a little bit, then the degree of belief in $A \,\&\, B$ must also change slightly. In other words, $f_{\&}$ must be *continuous.* So, we arrive at the following definitions:

Definition. *By an* AND-operation*, we mean a commutative, associative, monotonic, continuous operation* $f_{\&} : [0, 1] \times [0, 1] \to [0, 1]$ *with the properties* $f_{\&}(1, a) = a$ *and* $f_{\&}(0, a) = 0$.

Definition. *By an* OR-operation*, we mean a commutative, associative, monotonic, continuous operation* $f_{\vee} : [0, 1] \times [0, 1] \to [0, 1]$ *with the properties* $f_{\vee}(1, a) = 1$ *and* $f_{\vee}(0, a) = a$.

A complete classification of operations that satisfy these properties has been given in [36] (see also [1, 14, 44, 45]). The simplest AND and OR operations were first used for degrees of belief by L. Zadeh in [59]: $f_{\&}(x, y) = \min(x, y)$, $f_{\&}(x, y) = xy$, $f_{\vee}(x, y) = \max(x, y)$, and $f_{\vee}(x, y) = x + y - xy$. Since then, several other operations have been proposed.

One way to choose an operation is to choose many pairs of statements (A_k, B_k), ask experts to estimate $d(A_k)$, $d(B_k)$, and $d(A_k \,\&\, B_k)$ for every pair, and choose a function $f_{\&}$ for which $d(A_k \,\&\, B_k)$ is the closest to $f_{\&}(d(A_k), d(B_k))$ (e.g., in terms of least squares). This empirical approach was first implemented by the authors of the first successful expert system MYCIN [15, 47]. For different fields of expertise, different functions $f_{\&}$ emerge (for experimental results, see, e.g., [60]). This difference is easy to explain. If we know $a = d(A)$ and $b = d(B)$, then we can have more optimistic estimates for $d(A \,\&\, B)$ (e.g., $\min(a, b)$) and more cautious ones (e.g., $f_{\&}(a, b) = a \cdot b$). In some applications (e.g., in medicine), mistakes can be deadly, so more cautious estimates are needed. In other applications (e.g., in geology), we cannot measure as many parameters as in medicine, so, we have to rely more on expertise, and hence, experts must take risks. In these applications, more optimistic estimates are used: e.g., a geologist starts to drill in the uncertainty in which a surgeon is not likely to start an incision.

Similarly, NOT operations $f_\neg : [0,1] \to [0,1]$ are defined. A usual choice is $f_\neg(x) = 1-x$ (for degrees of belief from $[-1,1]$, it is $x \to -x$).

1.6 From AND and OR Operations to Degrees of Belief in Composite Statements

If we fix AND, OR, and NOT operations, then we can in principle, knowing the degree of belief in the basic statements, determine the degree of belief in their logical combination Q. To do that, we represent the given formula Q as a combination of &, $\vee$, and $\neg$, and then consequently use our chosen operations with degrees of belief instead of these logical symbols.

Definition. *Let V be a finite set. Its elements will be called* Boolean (propositional) variables. *By a propositional formula, we mean an arbitrary expression obtained from Boolean variables by using the symbols &, $\vee$, and $\neg$.*

Definition. *Let us fix an AND-operation $f_\&$, an OR-operation $f_\vee$, and a NOT operation $f_\neg$. For every propositional formula F, and for all numbers $a_1, ..., a_n$, we define $p^*_F(a_1, ..., a_N)$ as follows:*

- *If F coincides with the variable A_i, then $p^*_F = a_i$.*
- *If $F = F_1 \& F_2$, then $p^*_F = f_\&(p^*_{F_1}, p^*_{F_2})$.*
- *If $F = F_1 \vee F_2$, then $p^*_F = f_\vee(p^*_{F_1}, p^*_{F_2})$.*
- *If $F = \neg F_1$, then $p^*_F = f_\neg(p^*_{F_1})$.*

For example, a formula Q of the type $A \to B$ can be represented as $B \vee \neg A$, and therefore, as a degree of belief in Q, we can take $f_\vee(d(B), f_\neg(d(A))$. So, if $d(A) = 0.6$, $d(B) = 0.7$, and we use min, max, and $x \to 1-x$ for &, $\vee$, and $\neg$, then we get $d(Q) = \max(d(B), 1-d(A)) = \max(0.7, 1-0.6) = 0.7$.

1.7 The Need to Use Intervals

We have already mentioned that for every propositional combination F of the original statements S_j, the actual probability $p(Q)$ depends not only on the probabilities $P(S_j)$, but also on whether these statements are independent or correlated.

In the traditional expert system approach, we provide a single numerical estimate for $p(Q)$. It is desirable to also determine the interval $\mathbf{p}(F)$ of possible values of $p(F)$.

1.8 Intervals Are More Adequate: Additional Argument

In an ideal situation, degrees of belief can be described by exact real numbers. In practice, the situation is more complicated, because experts cannot describe their degrees of belief precisely. A review of eliciting degrees of belief describes this problem.

- If an expert describes his degree of belief by selecting, e.g., 8 on a scale from 0 to 10, this does not mean that his degree of belief is exactly 0.8: if instead, we ask him to select on a scale from 0 to 9, then whatever he chooses, after dividing it by 9, we will never get 0.8. If an expert chooses a value 8 on a 0 to 10 scale, then the only thing that we know about the expert's degree of belief is that it is closer to 8 than to 7 or to 9, i.e., that this degree of belief belongs to the *interval* $[0.75, 0.85]$.

  Another possible source of interval uncertainty is when we have *several* experts, and their estimates differ. If, e.g., two equally good experts point to 7 and 8, then, if we are cautious, we would rather describe the resulting degree of belief as the interval $[0.7, 0.8]$ (or, in view of the above remark, as the interval $[0.65, 0.85]$) [22, 54, 56, 57].

- If we determine the degree of belief by polling, then the same argument shows that the resulting numbers are not precise: e.g., if 8 out of 10 experts voted for A, then we cannot say that the actual degree of belief is exactly 0.8, because, if we repeated this procedure with 9 experts, we will never get exactly 0.8. In this case, there are two other sources of uncertainty:

  - Picking experts is sort of a random procedure, so, the result of voting is a statistical estimate that is not precise (just like a statistical frequency estimate of probability). A better description will be to give an *interval* of possible values of $d(A)$.

  - The polling method of estimating the degree of belief is based on the assumption that an expert can always tell whether he believes in a given statement S or not. Then, we take the ratio $d(S) = N(S)/N$ of the number $N(S)$ of experts who believe in S to the total number N of experts as the desired estimate. For $\neg S$, we thus have $N(\neg S) = N - N(S)$, and so, $d(\neg S) = N(\neg S)/N = 1 - d(S)$. In reality, an expert is often unsure about S. In this case, instead of dividing the experts into two categories: those who believe in S and those who do not, we must divide them into *three* categories: those who believe that S is true (we will denote their number by $N(S)$), those who believe that S is false (we will denote their number by $N(\neg S)$), and those who do not have the definite opinion about S; there are $N - N(S) - N(\neg S)$ of them. In this situation, one number is not sufficient to describe the experts' degree of belief in S, we need at least two. There are two ways to describe it:

    - We can describe the degree of belief in S as $d(S) = N(S)/N$ and the degree of belief in $\neg S$ as $d(\neg S) = N(\neg S)/N$. These two numbers must satisfy the condition $d(S) + d(\neg S) \leq 1$. This description is known under three different names: *interval probability theory* [16, 18, 42], *intuitionistic fuzzy logic* [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 24] or *a vague set* [25]. (The reason for the word "intuitionistic" is that this logic is close to the original intuitionistic idea that the law of excluded middle is not always true.)

  - We can describe the degree of belief $d(S)$ in S and the degree of *plausibility* of S estimating as the fraction of experts who do not consider S impossible, i.e., as $pl(S) = 1 - d(\neg S)$. This representation is called *Dempster-Shafer formalism* [20, 46, 50, 58] (for efficient algorithms, see, e.g., [32]).

- If we have chosen $d(A)$ by comparing $\mathcal{A}$ with lotteries, then at some point, an expert will be unable to make a reasonable choice. For example, hardly anyone can make a meaningful choice saying that the alternative $\mathcal{A}$ is better than a lottery with, say 50.1% of winning, but worse than a lottery with a 50.2% of winning. If we use bisection and do not allow an expert to say "I do not know", then we are *forcing* the expert to make some decision. The resulting decisions are arbitrary, and thus, may be inconsistent: e.g., if three alternatives A, B, and C are more or less equivalent to an expert, but we force him to choose, then he may choose A from the pair (A, B), B from the pair (B, C), and C from the pair (C, A) (see, e.g., [30, 60]). A more realistic estimate would emerge if we allow the experts to say "I do not know" as a result of the comparison. In this case, at some point, we will reach an interval $[p^-, p^+]$ that definitely contains our degree of belief, but which cannot be narrowed (because for the values p from this interval, the expert cannot meaningfully compare $\mathcal{A}$ with a lottery $L(p)$).

So, to describe degrees of belief adequately, we must use *intervals* instead of real numbers.

1.9 Using Intervals in Expert Systems: Traditional Approach and Its Drawbacks

It is known that in general, the problem of finding the exact bounds for $p(F)$ is NP-hard; see, e.g., [43]. This problem is NP-hard even if all the probabilities $p(S_j)$ are equal to 1 – because it is equivalent to the propositional satisfiability problem, a known NP-hard problem.

Expert systems use technique similar to straightforward interval computations [29]. Namely, for simple formulas we know the corresponding probability bounds [52, 57]: if we know the bounds $[\underline{a}, \overline{a}]$ for $p(A)$ and $[\underline{b}, \overline{b}]$ for $p(B)$, then:

- $p(\neg A)$ is in the interval $[1 - \overline{a}, 1 - \underline{a}]$;
- $p(A \,\&\, B)$ is in the interval $[\max(\underline{a} + \underline{b} - 1, 0), \min(\overline{a}, \overline{b})]$;
- $p(A \vee B)$ is in the interval $[\max(\underline{a}, \underline{b}), \min(\overline{a} + \overline{b}, 1)]$.

In the general case, we parse F and replace each computation step with the corresponding probability operation.

For example, let $F = (A \,\&\, B) \vee (A \,\&\, \neg B)$ and $p(A) = p(B) = 0.6$. The compiler would start with $F_1 = A$ and $F_2 = B$, then it would compute $F_3 = \neg B$, $F_4 = F_1 \,\&\, F_2$, $F_5 = F_1 \,\&\, F_3$, and finally $F = F_4 \vee F_5$. Thus, according to the above procedure, we first find the bounds for $p(F_3) = p(\neg B)$, then for $p(F_4) = p(A \,\&\, B)$ and $p(F_5) = p(A \,\&\, \neg B)$, and finally, the bounds for $p(F)$. As a result, we get $p(\neg B) = 1 - 0.6 = 0.4$,

$$\mathbf{p}(A \,\&\, B) = [\max(0.6 + 0.6 - 1, 0), \min(0.6, 0.6)] = [0.2, 0.6],$$

$$\mathbf{p}(A \,\&\, \neg B) = [\max(0.6 + 0.4 - 1, 0), \min(0.6, 0.4)] = [0, 0.4],$$

$$\mathbf{p}(F) = [\max(0, 0.2), \min(0.4 + 0.6, 1)] = [0.2, 1.0].$$

In this problem, F is equivalent to A, so $p(F) = 0.6$. Thus, similarly to interval computations, we can see that the resulting interval contained excess width.

1.10 Formulation of the Problem

The main objective of the research is to develop techniques that will enable us to produce, for different queries, more accurate estimates for the corresponding intervals of $p(Q)$.

1.11 Structure of the Thesis

In Chapter 2, we describe the new techniques for handling uncertainty in expert systems, techniques which are motivated by affine and Taylor arithmetic. We start with describing interval computations – a situation where similar problems arise. We then describe affine and Taylor arithmetic and show how the ideas behind affine and Taylor arithmetic can be applied to expert systems. We end this chapter with the description of the corresponding algorithm; first we describe it on a few examples, then in the general case.

In Chapter 3, we present numerical examples of using the new algorithm. Chapter 4 covers the software implementation of the new algorithm. Conclusions are presented in Chapter 5.

Chapter 2

New Techniques for Handling Uncertainty in Expert Systems Motivated by Affine and Taylor Arithmetic

2.1 Interval Computations – A Situation Where Similar Problems Arise

Our main objective is to develop new more accurate techniques for estimating the range of possible values of $p(Q)$ for different queries Q. To design such techniques, we will use the experience of a situation where similar problems arise: namely, the situation of interval computations.

The main problem of interval computations is as follows: In many application areas, we are interested in the values of the quantity y that is difficult (or even impossible) to measure directly. To estimate the value of this quantity, we measure related easier–to–measure quantities $x_1, \ldots, x_n$, and then use the known relation $y = f(x_1, \ldots, x_n)$ between x_i and y to estimate y.

Measurements are never 100% accurate; as a result, the measured value $\widetilde{x}_i$ of the i-th quantity is, in general, different from the actual value x_i of this quantity. Because of this difference $\Delta\widetilde{x}_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i \neq 0$, the estimated value $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is, in general, different from the actual (unknown) value $y = f(x_1, \ldots, x_n)$.

Often, the only information that we have about the measurement error Δx_i is its upper bound Δ_i, for which $|\Delta x_i| \le \Delta_i$. In this case, when we get the measurement result $\widetilde{x}_i$, the only information that we have about the actual (unknown) value of the measured quantity x_i is that x_i is in the interval $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. In such situations, we must find the range of possible values of $y = f(x_1, \ldots, x_n)$ when $x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.

In other words, we arrive at the following problem:

- Given: n intervals

$$[\widetilde{x}_1 - \Delta_1, \widetilde{x}_1 + \Delta_1], \ldots, [\widetilde{x}_n - \Delta_n, \widetilde{x}_n + \Delta_n]$$

  and an algorithmically defined function $f(x_1, \ldots, x_n)$,

- to estimate the range for f, i.e., the interval

$$\{f(x_1, \ldots, x_n) \,|\, x_1 \in [\widetilde{x}_1 - \Delta_1, \widetilde{x}_1 + \Delta_1], \ldots, x_n \in [\widetilde{x}_n - \Delta_n, \widetilde{x}_n + \Delta_n]\}$$

This problem of computing this range is called the main problem of interval computations.

2.2 Straightforward Interval Computations: Algorithm, Example

For simple functions like $f(x_1, x_2) = x_1 + x_2$ or $f(x_1, x_2) = x_1 \cdot x_2$, we have explicit formulas for the corresponding range. For example; for $f(x_1, x_2) = x_1 + x_2$, the range is equal to:

$$[\underline{x}_1, \overline{x}_1] + [\underline{x}_2, \overline{x}_2] = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2].$$

For $f(x_1, x_2) = x_1 - x_2$, the range is equal to:

$$[\underline{x}_1, \overline{x}_1] - [\underline{x}_2, \overline{x}_2] = [\underline{x}_1 - \overline{x}_2, \overline{x}_1 - \underline{x}_2].$$

For $f(x_1, x_2) = x_1 \cdot x_2$, the range is equal to:

$$[\underline{x}_1, \overline{x}_1] \cdot [\underline{x}_2, \overline{x}_2] =$$

$$[\min(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \overline{x}_2, \overline{x}_1 \cdot \underline{x}_2, \overline{x}_1 \cdot \overline{x}_2), \max(\overline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \overline{x}_2, \overline{x}_1 \cdot \underline{x}_2, \overline{x}_1 \cdot \overline{x}_2)].$$

For $f(x_1) = 1/x_1$, the range is equal to:

$$1/[\underline{x}_1, \overline{x}_1] = [1/\overline{x}_1, 1/\underline{x}_1] \,\text{if}\, 0 \notin [\underline{x}_1, \overline{x}_1].$$

Finally, for $f(x_1, x_2) = x_1/x_2$, the range can be computed as:

$$[\underline{x}_1, \overline{x}_1]/[\underline{x}_2, \overline{x}_2] = [\underline{x}_1, \overline{x}_1] \cdot (1/[\underline{x}_2, \overline{x}_2]).$$

These formulas form *interval arithmetic.*

For a more complex function $f(x_1, \ldots, x_n)$, in general, computing the range is NP-hard [33]. We can however compute a reasonable enclosure for this range by using the following *straightforward interval computations* algorithm:

- Parse a function f, i.e., represent it as a sequence of elementary arithmetic operations.
- Replace each elementary arithmetic operation with the corresponding operation from interval arithmetic.

Let us illustrate this algorithm on a simple example of estimating the range of a function $f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $x \in [0, 1]$. A compiler parses this function into the following sequence of elementary operations. We start with the value $r_1 = x_1$, and then compute:

- $r_2 := 1 - r_1$;
- $y := r_1 \cdot r_2$.

Here, r_2 denotes the intermediate computation result. According to straightforward interval computations, we replace each operation by a corresponding operation of interval arithmetic:

- $\mathbf{r}_2 := 1 - \mathbf{r}_1$;
- $\mathbf{y} := \mathbf{r}_1 \cdot \mathbf{r}_2$.

Here,

$$\mathbf{r}_2 := 1 - \mathbf{r}_1 = [1,1] - [0,1] = [0,1],$$

and

$$\mathbf{y} := \mathbf{r}_1 \cdot \mathbf{r}_2 = [0,1] \cdot [0,1] =$$

$$[\min(0 \cdot 0, 0 \cdot 1, 1 \cdot 0, 1 \cdot 1), \max(0 \cdot 0, 0 \cdot 1, 1 \cdot 0, 1 \cdot 1) = [0,1].$$

How good is this estimate? For our simple function, it is easy to compute its actual range. Namely, this function is smooth (differentiable) and, according to calculus, a minimum and a maximum of a differentiable function on an interval is attained either at one of this interval's endpoints, or at a point inside this interval where the derivative of this function is equal to 0.

For the function $f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $\mathbf{x}_1 = [0,1]$, the derivative is equal to $f'(x_1) = 1 \cdot (1 - x_1) + x_1 \cdot x_1 \cdot (-1) = 1 - 2 \cdot x_1$. So, $f'(x_1) = 0$ when $x = 1/2$. Thus, to find the minimum and the maximum of the function $f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $[0,1]$, it is sufficient to compute the value of this function at three points: two endpoints (0 and 1) and the point $x_1 = 1/2$ where the derivative is equal to 0. The smallest of these three values if the lower endpoint y of the desired range $[\underline{y}, \overline{y}]$ and the largest of these three values if the upper endpoint of the range $[\underline{y}, \overline{y}]$. For $f(x_1) = x_1 \cdot (1 - x_1)$, $f(0) = f(1) = 0$ and $f(1/2) = 1/4$. So,

$$\underline{y} = \min(f(0), f(1), f(1/2)) = \min(0, 0, 1/4) = 0$$

and

$$\overline{y} = \max(f(0), f(1), f(1/2)) = \max(0, 0, 1/4) = 1/4.$$

So, the range $[\underline{y}, \overline{y}]$ of the function $f(x_1) = x_1 \cdot (1 - x_1)$ is equal to $[0, 1/4]$. We can see that the estimate $[0,1]$ provided by the straightforward interval computations indeed encloses the actual range: $[0, 1] \supseteq [0, 1/4]$. We can also see that this estimate is much wider than the actual range.

How can we improve the estimates?

2.3 Affine Arithmetic (a.k.a. Generalized Interval Computations)

Why was our estimate for the range of $f(x_1) = x_1 \cdot (1 - x_1)$ too wide? For $r_1 = x_1$, we had the exact range $\mathbf{x}_1 = [0, 1]$. For $r_2 = 1 - x_1$, we also had the exact range $\mathbf{r}_2 = [0, 1]$ of possible values. The excess width came from the fact that when we transformed the ranges of r_1 and r_2 into the range of $y = r_1 \cdot r_2$, we used the formula for interval multiplication, the formula that ignores the fact that r_1 and r_2 are actually related. To take this relation into account, researchers developed the notion of affine arithmetic, a.k.a. generalized interval computations; see, e.g., [19, 23, 27].

In the beginning, we start with n intervals $[\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n]$. When intervals come from measurements, then $\underline{x}_i = \widetilde{x}_i - \Delta_i$ and $\overline{x}_i = \widetilde{x}_i + \Delta_i$, where $\widetilde{x}_i$ is the measurement result and Δ_i is the guaranteed upper bound on the measurement error $\Delta x_i = \widetilde{x}_i - x_i$. Thus, the actual (unknown) value x_i of the measured quantity is equal to $x_i = \widetilde{x}_i - \Delta x_i$, where $|\Delta x_i| \le \Delta_i$.

When we have the endpoints $\underline{x}_i = \widetilde{x}_i - \Delta_i$ and $\overline{x}_i = \widetilde{x}_i + \Delta_i$ of the interval $[\underline{x}_i, \overline{x}_i]$, then we can reconstruct the value $\widetilde{x}_i$ as a midpoint $\widetilde{x}_i = \dfrac{\underline{x}_i + \overline{x}_i}{2}$ of this interval, and the value Δ_i as the half-width $\Delta_i = \dfrac{\overline{x}_i - \underline{x}_i}{2}$ of the corresponding interval.

When intervals do not come from measurements, we can still represent each value x_i from the corresponding interval $[\underline{x}_i, \overline{x}_i]$ by the same expression $x_i = \widetilde{x}_i - \Delta x_i$, where $\widetilde{x}_i$ is the interval's midpoint and $|\Delta x_i| \le \Delta_i$, where Δ_i is the interval's half-width.

In affine arithmetic, we start with expressions $x_i = \widetilde{x}_i - \Delta x_i$, where $\Delta x_1, \ldots, \Delta x_n$ are new variables, and for each intermediate result r_j, we not only keep the interval of its possible values, but we also keep the explicit expression for the dependence of r_j on Δx_i, as $r_j = r_{j0} + r_{j1} \cdot \Delta x_1 + \ldots + r_{jn} \cdot \Delta x_n + \mathbf{R}_j$, where the interval $\mathbf{R}_j$ contains non-linear terms in the dependence. Specifically, for each intermediate result r_j, we keep the coefficients $r_{j0}, r_{j1}, \ldots, r_{jn}$, and the endpoints of the interval $\mathbf{R}_j$.

We start with n variables $x_i = \widetilde{x}_i - \Delta x_i$. Each of these variables is represented as:

$$x_i = \widetilde{x}_i + 0 \cdot \Delta x_1 + \ldots + 0 \cdot \Delta x_{i-1} + (-1) \cdot \Delta x_i + 0 \cdot \Delta x_{i+1} + \ldots + 0 \cdot \Delta x_n + [0, 0],$$

i.e., the corresponding coefficients are $(\widetilde{x}_i, 0, \ldots, 0, -1, 0, \ldots, 0, [0, 0])$.

At every computational step, we then process such expressions instead of processing numbers (as in original computations) or intervals (as in straightforward interval computations).

For each arithmetic operation $a * b$, if we start with $a = a_0 + a_1 \cdot \Delta x_1 + \ldots + a_n \cdot \Delta x_n + \mathbf{A}$ and $b = b_0 + b_1 \cdot \Delta x_1 + \ldots + b_n \cdot \Delta x_n + \mathbf{B}$, we get the following formulas for $c = a * b$. For addition $c = a + b$, we have

$$(a_0 + a_1 \cdot \Delta x_1 + \ldots + a_n \cdot \Delta x_n + \mathbf{A}) + (b_0 + b_1 \cdot \Delta x_1 + \ldots + b_n \cdot \Delta x_n + \mathbf{B}) =$$

$$(a_0 + b_0) + (a_1 + b_1)\Delta x_1 + \ldots + (a_n + b_n)\Delta x_n + (\mathbf{A} + \mathbf{B}),$$

i.e., we have

$$c_0 = a_0 + b_0, c_1 = a_1 + b_1, \ldots, c_n = a_n + b_n, \mathbf{C} = \mathbf{A} + \mathbf{B}.$$

Similarly, for subtraction $c = a - b$, we have

$$(a_0 + a_1 \cdot \Delta x_1 + \ldots + a_n \cdot \Delta x_n + \mathbf{A}) - (b_0 + b_1 \cdot \Delta x_1 + \ldots + b_n \cdot \Delta x_n + \mathbf{B}) =$$

$$(a_0 - b_0) + (a_1 - b_1)\Delta x_1 + \ldots + (a_n - b_n)\Delta x_n + (\mathbf{A} - \mathbf{B}),$$

i.e., we have

$$c_0 = a_0 - b_0, c_1 = a_1 - b_1, \ldots, c_n = a_n - b_n, \mathbf{C} = \mathbf{A} - \mathbf{B}.$$

For multiplication $c = a \cdot b$, we have

$$(a_0 + a_1 \cdot \Delta x_1 + \ldots + a_n \cdot \Delta x_n + \mathbf{A}) \cdot (b_0 + b_1 \cdot \Delta x_1 + \ldots + b_n \cdot \Delta x_n + \mathbf{B}) =$$

$$a_0 \cdot b_0 + (a_0 \cdot (b_1 \cdot \Delta x_1 + \ldots + b_n \cdot \Delta x_n) + b_0 \cdot (a_1 \cdot \Delta x_1 + \ldots + a_n \cdot \Delta x_n))+$$

$$a_0 \cdot \mathbf{B} + b_0 \cdot \mathbf{A} + (a_1 \cdot \Delta x_1 + \ldots + a_n \cdot \Delta x_n + \mathbf{A}) \cdot (b_1 \cdot \Delta x_1 + \ldots + b_n \cdot \Delta x_n + \mathbf{B}) =$$

$$a_0 \cdot b_0 + (a_0 \cdot b_1 + a_1 \cdot b_0) \cdot \Delta x_1 + \ldots + (a_0 \cdot b_n + a_n \cdot b_0) \cdot \Delta x_n +$$

$$a_0 \cdot \mathbf{B} + b_0 \cdot \mathbf{A} + (a_1 \cdot \Delta x_1 + \ldots + a_n \cdot \Delta x_n + \mathbf{A}) \cdot (b_1 \cdot \Delta x_1 + \ldots + b_n \cdot \Delta x_n + \mathbf{B}).$$

The last term is approximated by the product of the corresponding intervals. Since $\Delta x_i \in [-\Delta_i, \Delta_i]$, the range of possible values for $a_1 \cdot \Delta x_1 + \ldots + a_n \cdot \Delta x_n$ is $[-\sum |a_i|\Delta_i, \sum |a_i|\Delta_i]$, and so:

$$c_0 = a_0 + b_0, c_1 = a_0 \cdot b_1 + a_1 \cdot b_0, \ldots, c_n = a_0 \cdot b_n + a_n \cdot b_0,$$

$$\mathbf{C} = a_0 \cdot b + a \cdot b_0 + \left(\left[-\sum |a_i|\Delta_i, \sum |a_i|\Delta_i\right] + \mathbf{A}\right) \cdot \left(\left[-\sum |b_i|\Delta_i, \sum |b_i|\Delta_i\right] + \mathbf{B}\right)$$

There are similar formulas for the inverse and the ratio a/b.

For a general function $f(x_1, \ldots, x_n)$, general interval computations algorithm for computing the range of f is as follows:

- First, we parse the function f.
- Then, starting with expressions $x_i = \widetilde{x}_i - \Delta x_i$, we follow the computation of f step-by-step, replacing each elementary arithmetic operation with the corresponding operation of generalized interval (affine) arithmetic.
- As a result, we get an expression $y = y_0 + y_1 \Delta x_1 + \ldots + y_n \Delta x_n + \mathbf{Y}$ for the result $y = f(x_1, \ldots, x_n)$ of data processing.
- Based on this expression, we compute the range of y as
$$y = y_0 + \left[-\sum_{i=1}^{n} |y_i|\Delta_i, \sum_{i=1}^{n} |y_i|\Delta_i\right] + \mathbf{Y}.$$

Let us illustrate this algorithm on the same simple example of estimating the range of a function $f(x_1) = x_1 \cdot (1 - x_1)$. Here,

$$r_1 := x_1;$$

$$r_2 := 1 - r_1;$$

$$y := r_1 \cdot r_2.$$

Here, the midpoint $\widetilde{x}_1$ of the interval $[0,1]$ is 0.5 and its half-width $\Delta_1 = \dfrac{\overline{x}_1 - \underline{x}_1}{2}$ is also 0.5. In accordance with the affine arithmetic algorithm, we start with the expression $x_1 = \widetilde{x}_1 - \Delta x_1$, i.e.,

$$x_1 = 0.5 + (-1) \cdot \Delta x_1 + [0,0].$$

Here, $a_0 = 0.5$, $a_1 = -1$, and $\mathbf{A} = [0,0]$.

In these terms, the value 1 is represented as $1 + 0 \cdot \Delta x_1 + [0,0]$, i.e., as $b_0 = 1$, $b_1 = 0$, and $\mathbf{B} = [0,0]$. Thus, for $r_2 = b - a = 1 - r_1$, we have

$$c_0 = b_0 - a_0 = 1 - 0.5 = 0.5,$$

$$c_1 = b_1 - a_1 = 0 - (-1) = 1,$$

and

$$\mathbf{C} = \mathbf{B} - \mathbf{A} = [0,0] - [0,0] = [0,0].$$

In other words, for r_2, we have the expression

$$r_2 = 0.5 + \Delta x_1 + [0,0]$$

corresponding to

$$r_{20} = 0.5, \quad r_{21} = 1, \quad \mathbf{R}_2 = [0,0].$$

Now, for

$$y := r_1 \cdot r_2 = (0.5 - \Delta x_1) \cdot (0.5 + \Delta x_1) + [0,0],$$

we get

$$y_0 = a_0 \cdot r_{20} = 0.5 \cdot 0.5 = 0.25;$$

$$y_1 = a_0 \cdot r_{21} + a_1 \cdot r_{20} = (0.5 \cdot 1) + ((-1) \cdot 0.5) = 0;$$

$$\mathbf{Y} = a_0 \cdot \mathbf{R}_2 + \mathbf{A} \cdot r_{20} + ([-|a_1|\Delta_1, |a_1|\Delta_1] + \mathbf{A}) \cdot ([-|r_{21}|\Delta_1, |r_{21}|\Delta_1] + \mathbf{R}_2) =$$

$$0.5 \cdot [0,0] + [0,0] \cdot 0.5 + ([-0.5,0.5] + [0,0]) \cdot ([-0.5,0.5] + [0,0] =$$

$$[0,0] + [-0.5,0.5] \cdot [-0.5,0.5] = [-0.25,0.25].$$

Hence, for $y = f(x_1)$, we get

$$y = y_0 + y_1 \cdot \Delta x_1 + \mathbf{Y} = 0.25 + [-0.25, 0.25].$$

So, the estimated range of y is

$$0.25 + [-0.25, 0.25] = [0, 0.5].$$

This range is still wider than the actual range $[0, 0.25]$, but it is twice narrower than the range $[0, 1]$ obtained by straightforward interval computations.

How can we further decrease the range?

2.4 Taylor Models

The main reason why our estimate is still too wide is that, while we took into account the linear part of the dependence on Δx_1, we did not account for non-linear dependence. To take this into consideration, researchers developed *Taylor methods* [38], where at each step, instead of representing r_j as a linear combination of Δx_i, we represent r_j as a polynomial of some order, k, with an interval remainder part containing terms of higher order. For example, if we use quadratic methods, then for each intermediate quantity r_j, we use the following expression:

$$r_j = r_{j0} + \sum_{i=1}^{n} r_{ji} \cdot \Delta x_i + \sum_{i=1}^{n} \sum_{k=1}^{n} r_{jk} \cdot \Delta x_i \cdot \Delta x_k + \mathbf{R_j},$$

i.e., we represent each r_j as a sequence of values

$$r_{j0}, r_{j1}, \ldots, r_{jn}, r_{j11}, \ldots, r_{j1n}, r_{j21}, \ldots, r_{j2n}, \ldots, r_{jn1}, \ldots, r_{jnn},$$

and the endpoints of the interval $\mathbf{R}_j$.

Similarly to affine arithmetic, we perform arithmetic operations with these expressions and then include all higher order terms into the corresponding remainders. As a result, we get better and better estimates for the range at the expense of longer and longer computations:

- for straightforward interval computations, instead of each operation with real numbers, we have at most four operations with interval endpoints;
- for affine arithmetic, each original arithmetic operation translates into $O(n)$ operations with numbers; e.g., $c_i = a_i + b_i$, $0 \leq i \leq n$ for addition;
- for k-th order Taylor arithmetic, each original arithmetic operation translates into $O(n^k)$ operations with $O(n^k)$ numbers $a_0, a_i, a_{ij}, \ldots, a_{i_1,\ldots,i_k}$.

2.5 How to Apply the Main Ideas Behind Affine and Taylor Arithmetic to Uncertainty in Expert Systems

In Chapter 1, we have described a traditional approach for providing guaranteed bounds for the range of $p(Q)$ for different propositional combinations Q of the original statements $S_1, \ldots, S_n$. This approach is as follows:

- first, we parse the expression Q, i.e., represent it as a sequence of elementary propositional operations &, $\vee$, and $\neg$;
- next, we replace each elementary propositional operation by the corresponding operation with the intervals.

We can see that this algorithm is very similar to the algorithm of straightforward interval computations. So, to decrease the width of the corresponding range $\mathbf{p}(Q)$, it is reasonable to use the same ideas that we used to decrease the range in interval computations: namely, the ideas behind affine and taylor arithmetic.

In order to use these ideas, let us reformulate them in more general terms. In straightforward interval computations, at each intermediate step, we only keep the interval of possible values of the corresponding intermediate quantity r_j. In affine arithmetic, we also store the coefficients $r_{j1}, \ldots, r_{jn}$ that describe the dependence between r_j and the original variables $x_1, \ldots, x_n$. In some versions of affine arithmetic [51], when we take into account that every computational step adds round-off errors to the resulting uncertainty, we also keep the coefficients $r_{j,j-1}, r_{j,j-2}, \ldots,$ describing the relation between r_j and the previous intermediate results.

For processing uncertainty in expert systems, it is therefore reasonable, for each intermediate step, to describe not only the interval $\mathbf{p}(F_j)$ of possible values of $p(F_j)$, but also the possible values indicating possible dependence between F_j and each of the original statements $S_1, \ldots, S_n$ and previous intermediate results.

For Boolean variables F and S, the relation can be naturally described in terms of probabilities of different Boolean combinations of F and S, such as $p(F \,\&\, S), p(F \vee S)$, etc. For example:

- independence means that $p(F \,\&\, S) = p(F) \cdot p(S)$;
- the fact that F and S cannot happen at the same time would mean that $p(F \vee S) = p(F) + p(S)$;
- the fact that F and S are equivalent would mean that $p(F \,\&\, \neg S) = p(\neg F \,\&\, S) = 0$, etc.

Thus, to apply the ideas of affine arithmetic to handling uncertainty in expert systems, for each intermediate result F_j, besides an interval of possible values of

$$p(F_i \,\&\, F_j), p(F_i \vee F_j), \ldots,$$

for all previous intermediate expressions F_i and for all possible propositional functions of two variables.

2.6 Formulas for the Expert System Analogue of Affine Arithmetic

In the expert system analogue of affine arithmetic, on each step i, we not only compute the range of $p(R_i)$ but we also compute the range of probabilities for Boolean combinations of R_i and the previous values $R_j, j < i$.

In order to follow these computations, we must enumerate all possible Boolean combinations of two statements S and S'. Every such combination can be represented in the DNF form, as a union of basic combinations $S \,\&\, S'$, $S \,\&\, \neg S'$, $\neg S \,\&\, S'$, and $\neg S \,\&\, \neg S'$. First, we must find the ranges of the probabilities of such combinations: $p(S \,\&\, S')$, $p(S \,\&\, \neg S')$, $p(\neg S \,\&\, S')$, and $p(\neg S \,\&\, \neg S')$.

Next, we must find the ranges of probabilities for unions of two different basic combinations.

From four combinations, we can select six pairs. The union of two selected combinations is equal to the complement of the union of two others. Since $p(\neg A) = 1 - p(A)$, hence $\mathbf{p}(\neg A) = 1 - \mathbf{p}(A)$, it is sufficient to describe the intervals of probability for one of the two opposite combinations – we can always reconstruct the remaining one by subtracting the corresponding interval from 1.

In this text, we will use the following unions:

- the union of $S \,\&\, S'$ and $S \,\&\, \neg S'$, which is simply S;
- the union of $S \,\&\, S'$ and $\neg S \,\&\, S'$, which is simply S';
- the union of $S \,\&\, \neg S'$ and $\neg S \,\&\, S'$, which is a symmetric difference $S \Delta S'$.

The remaining three unions are complements to these ones:

- the union of and $\neg S \,\&\, \neg S'$, which is $\neg S$ – a complement of S;

- the union of $S \,\&\, \neg S'$ and $\neg S \,\&\, \neg S'$, which is $\neg S'$ – a complement to S';

- the union of $S \,\&\, S'$ and $\neg S \,\&\, \neg S'$, which is $S \equiv S'$ – a complement to $S \Delta S'$.

Every union of three basic combinations is a complement to the remaining combinations, so we do not need to separately compute the corresponding probability. For example:

- the union of $S \,\&\, S'$, $S \,\&\, \neg S'$, and $\neg S \,\&\, S'$ is $S \vee S'$ – which is a complement to $\neg S \,\&\, \neg S'$; actually, we will use $p(S \vee S')$ instead of $p(\neg S \,\&\, \neg S') = 1 - p(S \vee S')$;

- the union of $S \,\&\, S'$, $S \,\&\, \neg S'$, and $\neg S \,\&\, \neg S'$ is $S \vee \neg S'$ – a complement to $\neg S \,\&\, S'$;

- the union of $S \,\&\, S'$, $\neg S \,\&\, S'$, and $\neg S \,\&\, \neg S'$ is $S \vee S'$ – a complement to $S \,\&\, \neg S'$;

- the union of $\neg S \,\&\, S'$, $S \,\&\, \neg S'$, and $\neg S \,\&\, \neg S'$ is $S \,\&\, \neg S'$.

Finally, the union of all four basic combinations is a true statement whose probability is always 1.

So, in this approach, for every two statements S and S', in addition to estimating the range of probabilities $p(S)$ and $p(S')$ for S and S' for S and S', we also estimate the range of probabilities for the following Boolean combinations of S and S': $p(S \,\&\, S')$, $p(S \,\&\, \neg S')$, $p(\neg S \,\&\, S')$, $p(S \vee S')$, and $p(S \Delta S')$.

We already know how to compute the range of $p(S \,\&\, S')$ and $p(S \vee S')$. Let us know explain how we can do it for $S \Delta S'$.

For $R_1 \Delta R_2$, we know that

$$p(R_1 \Delta R_2) = p(R_1) + p(R_2) - 2 \cdot p(R_1 \,\&\, R_2),$$

so the largest possible value of $p(R_1 \Delta R_2)$ is when we have the smallest possible value of $p(R_1 \,\&\, R_2)$, i.e., $\min(p_1 + p_2 - 1, 0)$. Hence, the possible value of $p(R_1 \Delta R_2)$ is equal to $p_1 + p_2 - 2 \cdot \min(p_1 + p_2 - 1, 0)$.

Let us simplify this expression. We know that changing the sign changes the order of the numbers, i.e.,

$$-\min(a, b) = \max(-a, -b).$$

We also know that adding a number c to both sides does not change the order, so

$$c + \max(a, b) = \max(c + a, c + b),$$

Similarly, multiplying both sides by a positive number ($c > 0$) does not change the order, so

$$c \cdot \min(a, b) = \min(c \cdot a, c \cdot b).$$

In our case,

$$2 \cdot \min(p_1 + p_2 - 1, 0) = \min(2 \cdot (p_1 + p_2) - 1, 0),$$

hence,

$$-2 \cdot \min(p_1 + p_2 - 1, 0) = -\min(2 \cdot (p_1 + p_2) - 1, 0) =$$

$$\max(-2 \cdot (p_1 + p_2) - 1, 0) = \max(-2 \cdot -2 \cdot (p_1 + p_2), 0),$$

so,

$$p_1 + p_2 - 2 \cdot \min(p_1 + p_2 - 1, 0) = p_1 + p_2 + \max(2 - 2 \cdot (p_1 + p_2), 0) =$$

$$\max(p_1 + p_2 - 2 \cdot (p_1 + p_2), p_1 + p_2) = \max(2 - p_1 - p_2, p_1 + p_2).$$

Hence, if we know the probability p_1 of a statement R_1, and the probability p_2 of the statement R_2, then the largest possible value of $p(R_1 \Delta R_2)$ is equal to

$$\max(2 - p_1 - p_2, p_1 + p_2).$$

Similarly, the smallest possible value of $\underline{p}(R_1 \Delta R_2)$ is attained when we have the largest possible value of $p(R_1 \,\&\, R_2)$, i.e., $\min(p_1, p_2)$. Hence, this smallest value is equal to

$$p_1 + p_2 - 2 \cdot \min(p_1, p_2).$$

Similarly to the case of $\overline{p}(R_1 \Delta R_2)$, we conclude that

$$2 \cdot \min(p_1, p_2) = \min(2 \cdot p_1, 2 \cdot p_2),$$

hence,

$$-2 \cdot \min(p_1, p_2) = -\min(2 \cdot p_1, 2 \cdot p_2) = \max(-2 \cdot p_1, -2 \cdot p_2),$$

and

$$p_1 + p_2 - 2 \cdot \min(p_1, p_2) = p_1 + p_2 + \max(-2 \cdot p_1, -2 \cdot p_2) =$$

$$\max(p_1 + p_2 - 2 \cdot p_1, p_1 + p_2 - 2 \cdot p_2) = \max(p_2 - p_1, p_1 - p_2).$$

i.e., $\overline{p}(R_1 \Delta R_2) = |p_1 - p_2|$.

In other words, if we know the probabilities $p_1 = p(R_1)$ and $p_2 = p(R_2)$, then the range $\mathbf{p}(R_1 \Delta R_2)$ of possible values of $p(R_1 \Delta R_2)$ is

$$[|p_1 - p_2|, (2 - p_1 - p_2, p_1 + p_2)].$$

If, instead of the exact values of $p(R_1)$ and $p(R_2)$, we only know the intervals $[\underline{p}_1, \overline{p}_1]$ and $[\underline{p}_2, \overline{p}_2]$ of possible values of $p(R_1)$ and $p(R_2)$, then, to estimate $p(R_1 \Delta R_2)$, we must find the largest possible value of

$$\min(2 - p_1 - p_2, p_1 + p_2)$$

when $p_1 \in [\underline{p}_1, \overline{p}_1]$ and $p_2 \in [\underline{p}_2, \overline{p}_2]$ and to estimate $\underline{\mathbf{p}}(R_1 \Delta R_2)$, we must find the smallest possible value of $(p_1 - p_2)$ when $p_1 \in [\underline{p}_1, \overline{p}_1]$ and $p_2 \in [\underline{p}_2, \overline{p}_2]$.

Let us start with the bound $\overline{p}(R_1 \Delta R_2)$. The value

$$\min(2 - (p_1 + p_2), p_1 + p_2)$$

depends only on $p_1 + p_2$. It increases from 0 to 1 when $p_1 + p_2$ goes from 0 to 1, and then it decreases from 1 to 0 when $p_1 + p_2$ goes from 1 to its largest possible value 2.

When $p_1 \in [\underline{p}_1, \overline{p}_1]$ and $p_2 \in [\underline{p}_2, \overline{p}_2]$, then the interval of possible values of $p_1 + p_2$ is

$$[\underline{p}_1, \overline{p}_1] + [\underline{p}_2, \overline{p}_2] = [\underline{p}_1 + \underline{p}_2, \overline{p}_1 + \overline{p}_2].$$

If

$$\underline{p}_1 + \underline{p}_2 \le 1 \le \overline{p}_1 + \overline{p}_2,$$

then the largest possible value of this expression is attained at $p_1 + p_2 = 1$, and it is equal to 1.

If $\overline{p}_1 + \overline{p}_2 \le 1$, then we always have $p_1 + p_2 \le 1$, so

$$\min(2 - (p_1 + p_2), p_1 + p_2) = p_1 + p_2,$$

and the largest value of this expression is attained when $p_1 + p_2$ is the largest, i.e., when $p_1 + p_2 = \overline{p}_1 + \overline{p}_2$; this largest value is $\overline{p}_2 + \overline{p}_2$.

If $\underline{p}_1 + \underline{p}_2 \ge 1$, then we always have $p_1 + p_2 \ge 1$, so

$$\min(2 - (p_1 + p_2), p_1 + p_2) = 2 - (p_1 + p_2),$$

and the largest possible value of this expression is attained when $p_1 + p_2$ is the smallest, i.e., when $p_1 + p_2 = \underline{p}_1 + \underline{p}_2$; this largest value is equal to $2 - (\underline{p}_1 + \underline{p}_2)$. So:

- if $\underline{p}_1 + \underline{p}_2 \le 1 \le \overline{p}_1 + \overline{p}_2$, then $\overline{p}(R_1 \Delta R_2) = 1$;
- if $\overline{p}_1 + \overline{p}_2 < 1$, then $\overline{p}(R_1 \Delta R_2) = \overline{p}_1 + \overline{p}_2$;
- if $\underline{p}_1 + \underline{p}_2 > 1$, then $\overline{p}(R_1 \Delta R_2) = 2 - (\underline{p}_1 + \underline{p}_2)$.

One can check that we can combine these three cases into a single formula:

$$\overline{p}(R_1 \Delta R_2) = \min(\overline{p}_1 + \overline{p}_2, 2 - (\underline{p}_1 + \underline{p}_2), 1).$$

Indeed:

- if $\underline{p}_1 + \underline{p}_2 \le 1$ and $\overline{p}_1 + \overline{p}_2 \ge 1$, then $\overline{p}_1 + \overline{p}_2 \ge 1$, and $2 - (p_1 + p_2) \ge 1$ so 1 is indeed the smallest of these three values;
- if $\overline{p}_1 + \overline{p}_2 < 1$, then $\underline{p}_1 + \underline{p}_2 \le \overline{p}_1 + \overline{p}_2 < 1$, hence $2 - (p_1 + p_2) > 2 - 1 = 1$; in this case $\overline{p}_1 + \overline{p}_2$ is the smallest of the three values;
- if $\underline{p}_1 + \underline{p}_2 > 1$, then $\overline{p}_1 + \overline{p}_2 \ge \underline{p}_1 + \underline{p}_2 > 1$, hence $2 - (p_1 + p_2)$ is the smallest of the three values.

Similarly, the value

$$|p_1 - p_2| = \max(p_1 - p_2, p_2 - p_1)$$

depends on the difference $p_1 - p_2$. When $p_1 \in [0,1]$ and $p_2 \in [0,1]$, the difference $p_1 - p_2$ takes possible values from -1 to 1. When $p_1 - p_2$ increases from -1 to 0, the value $|p_1 - p_2|$ decreases from 1 to 0; when $p_1 - p_2$ increases from 0 to 1, the value $|p_1 - p_2|$ decreases from 0 to 1.

When $p_1 \in [\underline{p}_1, \overline{p}_1]$ and $p_2 \in [\underline{p}_2, \overline{p}_2]$, then the interval of possible values of $p_1 - p_2$ is

$$\left[\underline{p}_1, \overline{p}_1\right] - \left[\underline{p}_2, \overline{p}_2\right] = \left[\underline{p}_1 - \overline{p}_2, \overline{p}_1, \underline{p}_2\right].$$

If 0 is a possible value i.e., if $\underline{p}_1 - \overline{p}_2 \leq 0 \leq \overline{p}_1 - \underline{p}_2$, then the smallest possible value of $|p_1 - p_2|$ is 0.

If $\underline{p}_1 - \overline{p}_2 \geq 0$, this means that $p_1 - p_2$ is always greater than 0, so $|p_1 - p_2|$ attains its smallest possible value when $p_1 - p_2$ is the smallest, i.e., when $p_1 - p_2 = \underline{p}_1 - \overline{p}_2$; this smallest value is equal to $\underline{p}_1 - \overline{p}_2$.

If $\underline{p}_1 - \overline{p}_2 \leq 0$, this means that $p_1 - p_2$ is always negative. So, $|p_1 - p_2|$ attains its smallest value when $p_1 - p_2$ is the largest, i.e., when $p_1 - p_2 = \overline{p}_1 - \underline{p}_2$; this smallest value is equal to $|p_1 - p_2| = \underline{p}_2 - \overline{p}_1$.

Let us check that we can combine these three cases into a single expression:

$$\underline{p}(R_1 \Delta R_2) = \max(\underline{p}_1 - \overline{p}_2, \underline{p}_2 - \overline{p}_1, 0).$$

Indeed:

- if $\underline{p}_1 - \overline{p}_2 \leq 0 \leq \underline{p}_1 - \overline{p}_2$, then $\underline{p}_1 - \overline{p}_2 \leq 0$ and $\overline{p}_2 - \overline{p}_1 \leq 0$, so 0 is indeed the largest of these three expressions;

- if $\underline{p}_1 - \overline{p}_2 < 0$, then $\underline{p}_2 - \overline{p}_1 > 0$, so $\underline{p}_2 - \overline{p}_1$ is indeed the largest of these three expressions;

- if $\underline{p}_1 - \overline{p}_2 > 0$, then $\underline{p}_2 - \overline{p}_1 < 0$, so $\underline{p}_1 - \overline{p}_2$ is indeed the largest of these three expressions.

So,

$$\mathbf{p}(R_1 \Delta R_2) = [\underline{p}(R_1 \Delta R_2), \overline{p}(R_1 \Delta R_2)] = [\max(\underline{p}_1 - \overline{p}_2, \underline{p}_2 - \overline{p}_1, 0), \min(\overline{p}_1 + \overline{p}_2, 2 - (\underline{p}_1 + \underline{p}_2), 1)].$$

Let us illustrate this formula on a simple example. Let us assume that we know that $p_1 \in [0.2, 0.3]$ and $p_2 \in [0.4, 0.5]$, and we need to find the interval of possible values of $p(R_1 \Delta R_2)$. According to the formulas:

$$\underline{p}(R_1 \Delta R_2) = \max(\underline{p}_1 - \overline{p}_2, \underline{p}_2 - \overline{p}_1, 0),$$

and

$$\overline{p}(R_1 \Delta R_2) = \min(\overline{p}_1 + \overline{p}_2, 2 - (\underline{p}_1 + \underline{p}_2), 1).$$

Here,

$$\underline{p}(R_1 \Delta R_2) = \max(0.2 - 0.5, 0.4 - 0.3, 0) = 0.1,$$

and

$$\overline{p}(R_1 \Delta R_2) = \min(0.3 + 0.5, 2 - (0.2 + 0.3), 1) = 0.8.$$

Hence, $p(R_1 \Delta R_2) \in [0.1, 0.8]$.

2.7 Example of Applying the New Idea: $A \vee \neg A$

Let us start with the simple case when we know the probability $a = 0.6$ of a statement A, and we are interested in computing the probability of a propositional combination $Q \stackrel{\text{def}}{=} A \vee \neg A$.

From the commonsense viewpoint, Q is always true, so we should get $p(Q) = 1$. Let us find out what we will get if we apply the interval version of the traditional expert system approach.

According to this approach, first, we parse the expression $A \vee \neg A$. As a result, we get the following sequence of elementary propositional operations. We start with $R_1 = A$, then compute:

- $R_2 := \neg A = \neg R_1$;

- $Q := A \vee R_2 = R_1 \vee R_2$.

In accordance with the expert system approach, we start with $p(A) = 0.6 = [0.6, 0.6]$. Then, we perform, one by one, the interval versions of the corresponding propositional operations:

- first, we compute $p(R_2)$ as $p(R_2) = 1 - p(R_1) = 1 - 0.6 = 0.4$;

- next, we compute $p(Q)$ as $p(R_1 \vee R_2) = [0.4, 1]$.

Thus, we get excess width.

Let us now apply the new approach to the same problem. In the beginning, we still have only one statement $R_1 = A$, with probability $p(R_1) = 0.6$.

At the next step, when we consider $R_2 = \neg R_1$, we not only compute the range of probabilities for R_1, we also compute the range of possible values of the probabilities of all possible boolean combinations of R_1 and R_2. In particular, one of such combinations is $R_1 \vee R_2$. Since $R_2 = \neg R_1$, the probability of this boolean combination is 1.

Another combination is $R_1 \,\&\, R_2$. Since $R_2 = \neg R_1$, the probability of this boolean combination is 0. So, after we have performed the first step, we know not only that $p(R_2) = 0.4$, we also know that $p(R_1 \,\&\, R_2) = 0$, $p(R_1 \vee R_2) = 1$, etc.

On the next step, when we compute the probability of $Q \stackrel{\text{def}}{=} R_1 \vee R_2$, we can use all this information. Since we already know that $p(R_1 \vee R_2) = 1$, we thus conclude that $p(Q) = 1$.

So, instead of the interval $[0.4, 1]$ with excess width, we get the exact value 1.

2.8 Example of Applying the New Idea: $(A \,\&\, B) \vee (A \,\&\, \neg B)$

Let us now describe a slightly more complex example, when we know the probability $a = 0.6$ of a statement A, and the probability $b = 0.7$ of a statement B, and we are interested in computing the probability of a propositional combination $Q \stackrel{\text{def}}{=} (A \,\&\, B) \vee (A \,\&\, \neg B)$.

From the commonsense viewpoint, Q is equivalent to A, so we should get $p(Q) = 0.6$. Let us find out what we will get if we apply the interval version of the traditional expert system approach.

According to this approach, first, we parse the expression $(A \,\&\, B) \vee (A \,\&\, \neg B)$. As a result, we get the following sequence of elementary propositional operations. We start with $R_1 = A$ and $R_2 = B$, then compute:

- $R_3 := A \,\&\, B = R_1 \,\&\, R_2$;
- $R_4 := \neg B = \neg R_2$;
- $R_5 := A \,\&\, \neg B = R_1 \,\&\, R_4$;
- $Q := R_3 \vee R_5$.

Substituting values of the probabilities, we get $p(R_1) = 0.6 = [0.6, 0.6]$ and $p(R_2) = 0.7 = [0.7, 0.7]$. Then, we perform, one by one, the interval versions of the corresponding propositional operations. So, in accordance with traditional expert system approach, we compute the ranges of the corresponding Boolean combinations.

- first, we compute $p(R_3)$ as $p(R_3) = p(R_1 \,\&\, R_2) = [0.3, 0.6]$;
- next, we compute $p(R_4)$ as $p(R_4) = p(\neg R_2) = [0.3, 0.3]$;
- next, we compute $p(R_5)$ as $p(R_1 \,\&\, \neg R_2) = p(R_1 \,\&\, R_4) = [0, 0.3]$;
- next, we compute $p(Q)$ as $p(R_3 \vee R_5) = [0.3, 1]$.

Thus, we get excess width.

Let us now explain how this will be done for our new approach. At first, we have a value $p(R_1) = 0.6$ for $R_1 = A$. On the next step, we add another statement $R_2 = B$ with $p(R_2) = 0.7$. Hence, we get:

$$p(\neg R_1) = 1 - p(R_1) = 1 - 0.6 = 0.4$$

and

$$p(\neg R_2) = 1 - p(R_2) = 1 - 0.7 = 0.3.$$

To compute the range of $p(R_3) = p(R_1 \,\&\, R_2)$, we use the known formulas for the ranges of $p(S \,\&\, S^{'})$, $p(S \vee S^{'})$, and $p(S \,\Delta\, S^{'})$ to estimate the ranges for $R_1 \,\&\, R_2$, $R_1 \,\&\, \neg R_2$, $\neg R_1 \,\&\, R_2$, $R_1 \vee R_2$ and $R_1 \,\Delta\, R_2$:

$$\mathbf{p}(R_1 \,\&\, R_2) = [\max(p(R_1) + p(R_2) - 1, 0), \min(p(R_1), p(R_2))] =$$

$$[\max(0.6 + 0.7 - 1, 0), \min(0.6, 0.7)] = [0.3, 0.6];$$

$$\mathbf{p}(R_1 \,\&\, \neg R_2) = [\max(0.6 + 0.3 - 1, 0), \min(0.6, 0.3)] = [0, 0.3];$$

$$\mathbf{p}(\neg R_1 \,\&\, R_2) = [\max(0.4 + 0.7 - 1, 0), \min(0.4, 0.7)] = [0.1, 0.4];$$

$$\mathbf{p}(R_1 \vee R_2) = [\max(p(R_1), p(R_2), \min(p(R_1) + p(R_2) + 1)] =$$

$$[\max(0.6, 0.7), \min(0.6 + 0.7, 1)] = [0.7, 1];$$

$$\mathbf{p}(R_1 \,\Delta\, R_2) = \max(\underline{p}_1 - \overline{p}_2, \underline{p}_2 - \overline{p}_1, 0), \min(\overline{p}_1 + \overline{p}_2, 2 - (\underline{p}_1 + \underline{p}_2), 1)] =$$

$$[\max(-0.1, 0.1, 0), \min(1.3, 0.7, 1) = [0.1, 0.7].$$

On each step i, we not only compute the range of $p(R_i)$ but we also compute the range of probabilities for Boolean combinations of R_i and the previous statements $R_j, j < i$.

Hence, for $R_3 = R_1 \,\&\, R_2$, we not only compute the range of $p(R_3)$ but we also compute the range of probabilities for Boolean combinations of R_3 and the previous statements R_2 and R_1.

For that, we use the known relation $R_3 = R_1 \,\&\, R_2$ between the new statement R_3 and the previous statements R_1 and R_2, and the known ranges for $p(R_1 * R_2)$ for different Boolean combinations of R_1 and R_2.

Since $R_3 = R_1 \,\&\, R_2$, we conclude that

$$p(R_3) = p(R_1 \,\&\, R_2) = [0.3, 0.6].$$

For $R_3 \,\&\, R_1$, from the fact that $R_3 = R_1 \,\&\, R_2$, we conclude that

$$R_3 \,\&\, R_1 = R_1 \,\&\, R_2 \,\&\, R_1 = R_1 \,\&\, R_2,$$

therefore,

$$\mathbf{p}(R_3 \,\&\, R_1) = \mathbf{p}(R_1 \,\&\, R_2) = [0.3, 0.6].$$

For $R_3 \,\&\, \neg R_1$, from the fact that $R_3 = R_1 \,\&\, R_2$, we conclude that

$$R_3 \,\&\, \neg R_1 = R_1 \,\&\, R_2 \,\&\, \neg R_1,$$

i.e., $R_3 \,\&\, \neg R_1$ is impossible. The probability of an impossible event is 0, i.e.,

$$\mathbf{p}(R_3 \,\&\, \neg R_1) = [0, 0].$$

For $\neg R_3 \,\&\, R_1$, from the fact that $R_3 = R_1 \,\&\, R_2$, we conclude that

$$\neg R_3 \,\&\, R_1 = \neg(R_1 \,\&\, R_2) \,\&\, R_1 = R_1 \,\&\, \neg R_2,$$

hence,

$$\mathbf{p}(\neg R_3 \,\&\, R_1) = \mathbf{p}(R_1 \,\&\, \neg R_2) = [0, 0.3].$$

For $R_3 \vee R_1$, we have $R_3 \vee R_1 = R_1 \,\&\, R_2 \vee R_1 = R_1$, hence,

$$\mathbf{p}(R_3 \vee R_1) = \mathbf{p}(R_1) = [0.6, 0.6].$$

For $R_3 \,\Delta\, R_1$, we have

$$R_3 \,\Delta\, R_1 = (R_3 \,\&\, \neg R_1) \vee (\neg R_3 \,\&\, R_1) = R_1 \,\&\, \neg R_2,$$

so

$$\mathbf{p}(R_3 \,\Delta\, R_1) = \mathbf{p}(R_1 \,\&\, \neg R_2) = [0, 0.3].$$

Similarly, for combinations of R_3 and R_2, we have

$$\mathbf{p}(R_3 \,\&\, R_2) = \mathbf{p}(R_1 \,\&\, R_2) = [0.3, 0.6];$$

$$\mathbf{p}(R_3 \,\&\, \neg R_2) = [0, 0];$$

$$\mathbf{p}(\neg R_3 \,\&\, R_2) = \mathbf{p}(\neg R_1 \,\&\, R_2) = [0.1, 0.4];$$

$$\mathbf{p}(R_3 \vee R_2) = \mathbf{p}(R_2) = [0.7, 0.7];$$

$$\mathbf{p}(R_3 \,\Delta\, R_2) = \mathbf{p}(R_2 \,\&\, \neg R_1) = [0.1, 0.4].$$

Since, $R_4 = \neg R_2$, we have $p(R_4) = 1 - p(R_2) = 1 - 0.7 = 0.3$, the formulas for combinations of R_4 and R_1, R_3 can be easily deduced from the formulas for the combinations of R_2 with R_1, R_3. For example, $R_4 \,\&\, R_1 = \neg R_2 \,\&\, R_1$, so

$$\mathbf{p}(R_4 \,\&\, R_1) = \mathbf{p}(\neg R_2 \,\&\, R_1) = [0, 0.3].$$

Similarly,

$$\mathbf{p}(R_4 \,\&\, \neg R_1) = \mathbf{p}(\neg R_2 \,\&\, R_1) = 1 - \mathbf{p}(R_1 \vee R_2) = 1 - [0.7, 1] = [0, 0.3];$$

$$\mathbf{p}(\neg R_4 \,\&\, R_1) = \mathbf{p}(R_2 \,\&\, R_1) = [0.3, 0.6];$$

$$\mathbf{p}(R_4 \vee R_1) = \mathbf{p}(\neg R_2 \vee R_1) = 1 - \mathbf{p}(R_2 \,\&\, \neg R_1) = 1 - [0.1, 0.4] = [0.6, 0.9].$$

For $R_4 \,\Delta\, R_1$, we have

$$R_4 \,\Delta\, R_1 = \neg R_2 \,\Delta\, R_1 = (\neg R_2 \,\&\, \neg R_1) \vee (R_2 \,\&\, R_1),$$

i.e.,

$$R_4 \,\Delta\, R_1 = \neg(R_1 \,\Delta\, R_2),$$

hence,

$$\mathbf{p}(R_4 \,\Delta\, R_1) = 1 - \mathbf{p}(R_1 \,\Delta\, R_2) = 1 - [0.1, 0.7] = [0.3, 0.9].$$

Similarly,

$$\mathbf{p}(R_4 \,\&\, R_3) = \mathbf{p}(\neg R_2 \,\&\, R_3) = [0, 0];$$

$$\mathbf{p}(R_4 \,\&\, \neg R_3) = \mathbf{p}(\neg R_2 \,\&\, \neg R_3) = 1 - \mathbf{p}(R_2 \vee R_3) = 1 - [0.7, 0.7] = [0.3, 0.3];$$

$$\mathbf{p}(R_4 \vee R_3) = 1 - \mathbf{p}(R_2 \,\&\, \neg R_3) = 1 - [0.1, 0.4] = [0.6, 0.9];$$

$$\mathbf{p}(R_4 \,\Delta\, R_3) = 1 - \mathbf{p}(R_2 \,\Delta\, R_3) = 1 - [0.1, 0.4] = [0.6, 0.9].$$

Since $R_4 = \neg R_2$, the ranges of probabilities for propositional combinations of $R_4 * R_2$ are straightforward.

Since $R_4 = \neg R_2$, it is impossible to have $R_4 \,\&\, R_2$, so

$$\mathbf{p}(R_4 \,\&\, R_2) = [0, 0].$$

For $R_4 \,\&\, \neg R_2$, we get

$$R_4 \,\&\, \neg R_2 = \neg R_2,$$

so

$$\mathbf{p}(R_4 \,\&\, \neg R_2) = \mathbf{p}(\neg R_2) = [0.3, 0.3].$$

For $\neg R_4 \,\&\, R_2$, we get

$$\neg R_4 \,\&\, R_2 = R_2,$$

so

$$\mathbf{p}(\neg R_4 \,\&\, R_2) = \mathbf{p}(R_2) = [0.7, 0.7].$$

For $R_4 \vee R_2$, since $R_4 = \neg R_2$, disjunction is always true, so

$$\mathbf{p}(R_4 \vee R_2) = [1, 1].$$

Similarly,

$$R_4 \,\Delta\, R_2 = \neg R_2 \,\Delta\, R_2 = (\neg R_2 \,\&\, \neg R_2) \vee (R_2 \,\&\, R_2) = (\neg R_2 \vee R_2)$$

is always true, so

$$\mathbf{p}(R_4 \,\Delta\, R_2) = [1, 1].$$

We know that $R_5 = R_4 \,\&\, R_1$, so

$$\mathbf{p}(R_5) = \mathbf{p}(R_4 \,\&\, R_1) = [0, 0.3].$$

To find the ranges for probabilities $p(R_5 * R_i)$, we will use the known probability ranges and the relation $R_5 = R_1 \,\&\, R_4$.

For $R_5 \,\&\, R_1$, from the fact that $R_5 = R_1 \,\&\, R_4$, we conclude that

$$R_5 \,\&\, R_1 = R_1 \,\&\, R_4 \,\&\, R_1 = R_1 \,\&\, R_4,$$

so

$$\mathbf{p}(R_5 \,\&\, R_1) = \mathbf{p}(R_1 \,\&\, R_4) = [0, 0.3].$$

Similarly,

$$\mathbf{p}(R_5 \,\&\, \neg R_1) = [0, 0];$$

$$\mathbf{p}(\neg R_5 \,\&\, R_1) = \mathbf{p}(R_1 \,\&\, \neg R_4) = [0.3, 0.6];$$

$$\mathbf{p}(R_5 \vee R_1) = \mathbf{p}(R_1) = [0.6, 0.6];$$

$$\mathbf{p}(R_5 \,\Delta\, R_1) = \mathbf{p}(R_1 \,\&\, R_4) = [0.3, 0.6].$$

For combining R_5 and R_4, we have similar formulas:

$$\mathbf{p}(R_5 \,\&\, R_4) = [0, 0.3];$$

$$\mathbf{p}(R_5 \,\&\, \neg R_4) = [0, 0];$$

$$\mathbf{p}(\neg R_5 \,\&\, R_4) = \mathbf{p}(\neg R_1 \,\&\, R_4) = [0, 0.3];$$

$$\mathbf{p}(R_5 \vee R_4) = \mathbf{p}(R_4) = [0.3, 0.3];$$

$$\mathbf{p}(R_5 \,\Delta\, R_4) = \mathbf{p}(R_4 \,\&\, \neg R_1) = [0, 0.3].$$

For combining R_5 and R_2, we can use the previously known ranges. For example, we know that $R_5 = R_1 \,\&\, R_4$, and we know that R_4 and R_2 are incompatible $p(R_2 \,\&\, R_4) = 0$, so R_5 and R_2 are also incompatible:

$$\mathbf{p}(R_5 \,\&\, R_2) = [0, 0].$$

For $R_5 \,\&\, \neg R_2$, we know that $p(R_2 \,\Delta\, R_4) = 1$, which means that R_4 is exactly the complement to R_2, so $R_5 \,\&\, R_2 = R_5 \,\&\, R_4$, and

$$\mathbf{p}(R_5 \,\&\, \neg R_2) = \mathbf{p}(R_5 \,\&\, R_4) = [0, 0.3].$$

Comment. Please note that on this stage, the only information we use is the relation between R_5 and the previous statements, R_i, and the probability ranges for the Boolean combinations of the previous statements R_i. We can no longer use the logical relation between these previous statements. So, we could not use the relation $R_4 = \neg R_2$, we had to deduce it from the known probability ranges.

For $\neg R_5 \,\&\, R_2$, we similarly conclude that

$$\mathbf{p}(\neg R_5 \,\&\, R_2) = \mathbf{p}(\neg R_5 \,\&\, R_4) = 1 - \mathbf{p}(R_5 \vee R_4) = 1 - [0.3, 0.3] = [0.7, 0.7].$$

For $R_5 \vee R_2$, we conclude that

$$R_5 \vee R_2 = R_5 \vee \neg R_4 = \neg(\neg R_5 \,\&\, R_4),$$

so

$$\mathbf{p}(R_5 \vee R_2) = 1 - \mathbf{p}(\neg R_5 \,\&\, R_4) = 1 - [0, 0.3] = [0.7, 1].$$

For $R_5 \,\Delta\, R_2$, we conclude that

$$R_5 \,\Delta\, R_2 = R_5 \,\Delta\, \neg R_4 = (R_5 \,\&\, R_4) \vee (\neg R_5 \,\&\, \neg R_4) = \neg(R_5 \,\Delta\, R_4),$$

so

$$\mathbf{p}(R_5 \,\Delta\, R_2) = 1 - (R_5 \,\Delta\, R_4) = 1 - [0, 0.3] = [0.7, 1].$$

For $R_5 \,\&\, R_3$, we have $\mathbf{p}(R_3 \,\&\, \neg R_2) = [0, 0]$ and $\mathbf{p}(R_5 \,\&\, R_2) = [0, 0]$, so we conclude that R_3 only occurs when R_2 is true and R_5 only occurs when R_2 is false. Therefore, we cannot have both R_3 and R_5 true at the same time, i.e.,

$$\mathbf{p}(R_5 \,\&\, R_3) = [0, 0].$$

For $R_5 \,\&\, \neg R_3$, since R_5 and R_3 cannot happen together, we get $R_5 \,\&\, \neg R_3 = R_5$, so

$$\mathbf{p}(R_5 \,\&\, \neg R_3) = \mathbf{p}(R_5) = [0, 0.3].$$

For $\neg R_5 \,\&\, R_3$, we similarly have $\neg R_5 \,\&\, R_3 = R_3$, so

$$\mathbf{p}(\neg R_5 \,\&\, R_3) = \mathbf{p}(R_3) = [0.3, 0.6].$$

For $R_5 \vee R_3$, since R_3 and R_5 cannot happen together, we have

$$p(R_5 \vee R_3) = p(R_5) + p(R_3),$$

hence,

$$\mathbf{p}(R_5 \vee R_3) = \mathbf{p}(R_5) + \mathbf{p}(R_3) = [0, 0.3] + [0.3, 0.6] = [0.3, 0.9].$$

We also know that $\mathbf{p}(R_3 \,\&\, \neg R_1) = 0$ and $\mathbf{p}(R_5 \,\&\, \neg R_1) = 0$, which means that each R_3 and R_5 can only occur when R_1 is true. Thus, $R_5 \vee R_3$ can only occur if R_1 is true, so $p(R_5 \vee R_3) \le p(R_1) = 0.6$. Hence, out of the interval $[0.3, 0.9]$, only values $\in [0.3, 0.6]$ are possible:

$$\mathbf{p}(R_5 \vee R_3) = [0.3, 0.6].$$

For $R_5 \,\Delta\, R_3$, since R_3 and R_5 cannot occur together, we have $R_5 \,\Delta\, R_3 = R_5 \vee R_3$, so

$$\mathbf{p}(R_5 \,\Delta\, R_3) = \mathbf{p}(R_5 \vee R_3) = [0.3, 0.6].$$

Finally, we are interested in the probability range for $Q = R_3 \vee R_5$. We already know the range for $R_3 \vee R_5$, so we get

$$\mathbf{p}(Q) = [0.3, 0.6].$$

This estimate still contains the excess width in comparison to the correct value 0.6, but the resulting interval $[0.3, 0.6]$ is much narrower than the interval $[0.3, 1]$ that we obtained when we only estimated $\mathbf{p}(F_i)$.

To get an accurate estimate, we must consider probability ranges for triple combinations, etc.

2.9 Expert System Analogue of Affine and Taylor Arithmetic: Towards an Efficient Algorithm

2.9.1 Main Idea Behind the Proposed New Algorithm

In the above example, most computations were rather straightforward applications of the previously derived formulas. However, in some of these computations, especially when we computed bounds for $p(R_5 * R_3)$ for different propositional combinations $*$, we have to use a lot of reasoning to figure out how to combine known bounds $\mathbf{p}(R_i * R_j)$ into bounds for combinations involving the newly added statement R_k (in the above example, (R_5).

Our objective is to come up with a computationally efficient algorithm that would find the desired estimate without the need for any expert reasoning. To come up with such an algorithm, let us recall, e.g., how we estimated $\mathbf{p}(R_5 \vee R_3)$. First, we considered the fact that $\mathbf{p}(R_5 \,\&\, R_2) = [0,0]$ and $\mathbf{p}(R_3 \,\&\, \neg R_2) = [0,0]$ and concluded that $\mathbf{p}(R_5 \vee R_3) \in [0.3, 0.9]$. Next, we considered the fact that $\mathbf{p}(R_3 \,\&\, \neg R_1) = [0,0]$, and $\mathbf{p}(R_5 \,\&\, \neg R_1) = [0,0]$, and concluded that $p(R_5 \vee R_3) \in [0, 0.6]$. Then, as the estimated range for $\mathbf{p}(R_5 \vee R_3)$, we took the intersection

$$[0.3, 0.6] \cap [0, 0.6] = [0.3, 0.6]$$

of these two estimates. Let us transform this idea into the actual algorithm. We have already decided to select a parameter k indicating for which combinations $p(R_{i_1} * \ldots * R_{i_k})$ of the statements R_i we will estimate probability ranges:

- $k = 1$ means that we only estimate ranges for $\mathbf{p}(R_i)$ for the probabilities $p(R_i)$ – as in the original interval-valued expert system approach;
- $k = 2$ means that we also estimate ranges $\mathbf{p}(R_i * R_j)$ for propositional combinations of pairs R_i, R_j;
- $k = 3$ means that, in addition to probabilities of statements and combinations of pairs, we also estimate ranges $\mathbf{p}(R_i * R_j * R_k)$ for propositional combinations of triples, etc.

2.9.2 New Algorithm: Description

In addition to the parameter k, let us now introduce another parameter l such that, when estimating interval for $p(R_i * \ldots)$, we take into account only $\le l$ known probability bounds – and then take the intersection of the resulting estimate intervals corresponding to all possible sets of l probability bounds.

As a result, we arrive at the following algorithm. For each new statement $R_{j+1} = R_{i_1} * R_{i_2}$, where $*$ is any propositional operation, we must estimate the probabilities of all combinations $f(R_{j+1}, R_{j_1}, \ldots, R_{j_{k-1}})$, i.e., combinations

$$f(R_{i_1} * R_{i_2}, R_{j_1}, \ldots, R_{j_{k-1}})$$

involving $k + 1$ formulas R_i. We assume that we know the probability intervals for l such combinations of $\le k$ values R_i. Between these $l + 1$ propositional combinations, we involve $m \le (k + 1) + l \cdot k$ variables $R_{k_1}, \ldots, R_{k_m}$. We can describe both known and estimated probabilities as sums of probabilities of atomic statements $R_{i_1}^{\varepsilon_1} \,\&\, \ldots \,\&\, R_{i_m}^{\varepsilon_m}$, where $\varepsilon \in \{-, +\}$, R^+ means R, and R^- means $\neg R$. Then, we use linear programming (LP) to get desired bounds on the unknown probability.

In the following chapter, we give details on how to use LP and what are the results of using LP.

2.9.3 Computational Complexity of the New Algorithm

A propositional function $f(x_1, \ldots, x_k)$ of k propositional variables can be described as a function from the set $\{0, 1\}^k$ of 2^k possible combinations x_i to the set $\{0, 1\}$ of possible truth values. Thus, there are exactly 2^{2^k} such functions. For fixed k and l, this means that we have $O(1)$ such functions.

One the j-th step, we have j intermediate results $R_1, \ldots, R_j$. We have $O(j^k)$ possible combinations of $\le k$ such values, so we have $O(j^k)$ probability bounds. To compute each of $O(j^k)$ new bounds, we consider all possible subsets of l probabilities. There are $O((j^k)^l) =$

$O(j^{k \cdot l})$ such subsets. For each subset, for fixed k and l, the value m is bounded by a constant: $m = O(1)$. There are $2^m = O(1)$ possible combinations, so each LP requires $O(1)$ time. So, overall, on step j, we need $O(j^k) \cdot O(j^{k \cdot l}) \le M \cdot j^{k \cdot (l+1)}$ steps for some constant M.

Overall, we need $\le M(1^{k \cdot (l+1)} + \ldots + n^{k \cdot (l+1)})$ steps, where the number n of parsing steps is bounded by the length of the formula R. It is known that $1^a + 2^a + \ldots + n^a = O(n^{a+1})$, so overall, this algorithm requires $O(n^{k \cdot (l+1)+1})$ steps. In other words, the running time grows polynomially with the length of the formula R – so this algorithm is feasible [17].

Comment. It is worth mentioning that when $k \to \infty$ and $l \to \infty$, we get exact results; however, computation time grows exponentially with k and l, so we cannot realistically use too large values k and l.

Chapter 3

Using the New Algorithm: Numerical Examples

3.1 Estimating $p(A \vee B)$

Let us first illustrate the new algorithm described in Chapter 2 on the example when we already know the analytical solution: we know $p(A) = a = 0.6$ and $p(B) = b = 0.6$ and we want to estimate $p(A \vee B)$.

In accordance with the algorithm, we consider all $2^2 = 4$ combinations $A^{\varepsilon_1} \,\&\, B^{\varepsilon_2}$, where $\varepsilon \in \{-,+\}$, A^+ means A, and A^- means $\neg A$. In our case, we must consider 4 combinations $A \,\&\, B$, $A \,\&\, \neg B$, $\neg A \,\&\, B$, and $\neg A \,\&\, \neg B$. Let us use the following mnemonic notation to describe the (unknown) probabilities of these propositional combinations:

$$p_{++} = p(A \,\&\, B),$$

$$p_{+-} = p(A \,\&\, \neg B),$$

$$p_{-+} = p(\neg A \,\&\, B),$$

$$p_{--} = p(\neg A \,\&\, \neg B).$$

The four events $A \,\&\, B$, $A \,\&\, \neg B$, $\neg A \,\&\, B$, and $\neg A \,\&\, \neg B$ are disjoint and every situation belongs to one of them, so

$$p(A \,\&\, B) + p(A \,\&\, \neg B) + p(\neg A \,\&\, B) + p(\neg A \,\&\, \neg B) = 1.$$

In terms of our notations, this equality takes the form $p_{++} + p_{+-} + p_{-+} + p_{--} = 1$.

Here, $A = (A \,\&\, B) \vee (A \,\&\, \neg B)$, so

$$p(A) = p(A \,\&\, B) \vee p(A \,\&\, \neg B) = p_{++} + p_{+-}.$$

Since we know that $p(A) = a = 0.6$, we conclude that

$$p_{++} + p_{+-} = a = 0.6.$$

Similarly, $B = (A \,\&\, B) \vee (\neg A \,\&\, B)$, so

$$p(B) = p(A \,\&\, B) \vee p(\neg A \,\&\, \neg B) = p_{++} + p_{-+}.$$

Since we know that $p(B) = b = 0.6$, we conclude that

$$p_{++} + p_{-+} = 0.6.$$

Finally,

$$A \vee B \equiv (A \,\&\, B) \vee (A \,\&\, \neg B) \vee (\neg A \,\&\, B).$$

Since different disjunctions $A^{\varepsilon_1} \,\&\, B^{\varepsilon_2}$ cannot occur at the same time, we conclude that

$$p(A \vee B) = p(A \,\&\, B) + p(A \,\&\, \neg B) + p(\neg A \,\&\, B),$$

i.e., in our notations,

$$p(A \vee B) = p_{++} + p_{+-} + p_{-+}.$$

Thus, to find the largest possible value of $p(A \vee B)$, we must solve the following optimization problem:

Maximize $p_{++} + p_{+-} + p_{-+}$ under the constraints

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1;$$

$$p_{++} + p_{+-} = a;$$

$$p_{++} + p_{-+} = b;$$

$$p_{++} \geq 0; \quad p_{+-} \geq 0; \quad p_{-+} \geq 0; \quad p_{--} \geq 0.$$

Similarly, to find the smallest possible value of $p(A \vee B)$, we must solve the following optimization problem:

Minimize $p_{++} + p_{+-} + p_{-+}$ under the constraints

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1;$$

$$p_{++} + p_{+-} = a;$$

$$p_{++} + p_{-+} = b;$$

$$p_{++} \geq 0; \quad p_{+-} \geq 0; \quad p_{-+} \geq 0; \quad p_{--} \geq 0.$$

In both optimization problems, we optimize a linear function under constraints which are linear equalities and inequalities, so both problems are particular cases of a general linear programming problem.

In particular, in our case, when $a = b = 0.6$, we have the following two linear programming problems:

Maximize $p_{++} + p_{+-} + p_{-+}$ under the constraints

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1;$$

$$p_{++} + p_{+-} = 0.6;$$

$$p_{++} + p_{-+} = 0.6;$$

$$p_{++} \geq 0; \quad p_{+-} \geq 0; \quad p_{-+} \geq 0; \quad p_{--} \geq 0.$$

Minimize $p_{++} + p_{+-} + p_{-+}$ under the constraints

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1;$$

$$p_{++} + p_{+-} = 0.6;$$

$$p_{++} + p_{-+} = 0.6;$$

$$p_{++} \geq 0; \quad p_{+-} \geq 0; \quad p_{-+} \geq 0; \quad p_{--} \geq 0.$$

It is known that in general, the solution of a LP problem is attained at one of the vertices of the corresponding set, i.e., when the largest possible number of inequalities become equalities.

In our constraints, we have 3 linear equalities and 4 unknowns. In general, once we have as many equalities as unknowns, we can get a unique solution. So for 4 unknowns, we can have at most 4 linear equalities. Thus, we can have at most 1 inequality becoming an equality.

So, to find the minimum and the maximum of $p(A \vee B)$, we must consider 4 situations in which each of the 4 inequalities

$$p_{++} \geq 0; p_{+-} \geq 0; p_{-+} \geq 0; p_{--} \geq 0$$

becomes an equality. Let us consider these situations one by one.

If $p_{++} = 0$, then from $p_{++} + p_{+-} = 0.6$, we conclude that $p_{+-} = 0.6$, and from $p_{++} + p_{-+} = 0.6$, we conclude that $p_{-+} = 0.6$. So,

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 0 + 0.6 + 0.6 + p_{--} \geq 1.2 > 1.$$

This is inconsistent with the constraint

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1,$$

so, the case $p_{++} = 0$ is impossible.

If $p_{+-} = 0$, then from $p_{++} + p_{+-} = 0.6$, we conclude that $p_{++} = 0.6$. Now, from the constraint $p_{++} + p_{-+} = 0.6$, we conclude that

$$p_{-+} = 0.6 - p_{++} = 0.6 - 0.6 = 0.$$

Finally, from the constraint

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1,$$

we conclude that

$$p_{--} = 1 - p_{++} + p_{+-} + p_{-+} = 1 - 0.6 - 0 - 0 = 0.4.$$

In this case,

$$p(A \vee B) = p_{++} + p_{+-} + p_{-+} = 0.6 + 0 + 0 = 0.6.$$

If $p_{-+} = 0$, then from $p_{++} + p_{-+} = 0.6$, we conclude that $p_{++} = 0.6$. Now, from the constraint $p_{++} + p_{+-} = 0.6$, we conclude that

$$p_{+-} = 0.6 - p_{++} = 0.6 - 0.6 = 0.$$

From the constraint

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1,$$

we conclude that

$$p_{--} = 1 - p_{++} + p_{+-} + p_{-+} = 1 - 0.6 - 0 - 0 = 0.4.$$

In this case,

$$p(A \vee B) = p_{++} + p_{+-} + p_{-+} = 0.6 + 0 + 0 = 0.6.$$

Finally, if $p_{--} = 0$, then from

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1,$$

we conclude that

$$p_{++} + p_{+-} + p_{-+} = 1 - p_{--} = 1 - 0 = 1.$$

Since, $p_{++} + p_{+-} = 0.6$, we thus conclude that

$$p_{-+} = (p_{++} + p_{+-} + p_{-+}) - (p_{++} + p_{+-}) = 1 - 0.6 = 0.4.$$

Similarly, since $p_{++} + p_{-+} = 0.6$, we conclude that

$$p_{+-} = (p_{++} + p_{+-} + p_{-+}) - (p_{++} + p_{-+}) = 1 - 0.6 = 0.4.$$

Since, $p_{++} + p_{+-} = 0.6$ and $p_{+-} = 0.4$, we conclude that $p_{++} = 0.6 - 0.4 = 0.2$.
In this case,

$$p(A \vee B) = p_{++} + p_{+-} + p_{-+} = 0.2 + 0.4 + 0.4 = 1.$$

We know that the minimum and maximum of $p(A \vee B)$ is attained at one of the vertices. We have shown that at the vertices, $p(A \vee B)$ takes the values 0.6, 0.6 and 1. The smallest of these 3 values is 0.6, the largest of these 3 values is 1, so we conclude that the range of possible values of $p(A \vee B)$ is $[0.6, 1]$.

This conclusion is in perfect accordance with the formulas for the range $\mathbf{p}(A \vee B)$, according to which

$$\mathbf{p}(A \vee B) = [\max(a, b), \min(a + b, 1)] = [\max(0.6, 0.6), \min(0.6 + 0.6, 1)] = [0.6, 1].$$

3.2 Estimating the Probability Range for $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$: First Example

In the estimation presented in Chapter 2, when we estimated the probability $p(R_5 \,\&\, R_3)$, we used the formulas $\mathbf{p}(R_5 \,\&\, R_2) = [0, 0]$ and $\mathbf{p}(R_3 \,\&\, \neg R_2) = [0, 0]$ to conclude that $\mathbf{p}(R_5 \,\&\, R_3) = [0, 0]$.

In Chapter 2, to come up with this conclusion, we used some reasoning. Let us show that we can make the same conclusion by using linear programming.

Here, we have 3 basic statements used: R_5, R_3, and R_2, so we have $2^3 = 8$ possible combinations

$$R_5^{\varepsilon_5} \,\&\, R_3^{\varepsilon_3} \,\&\, R_2^{\varepsilon_2},$$

and thus, 8 unknowns:

$$p_{---} = p(\neg R_5 \,\&\, \neg R_3 \,\&\, \neg R_2),$$

$$p_{--+} = p(\neg R_5 \,\&\, \neg R_3 \,\&\, R_2),$$

$$p_{-+-} = p(\neg R_5 \,\&\, R_3 \,\&\, \neg R_2),$$

$$p_{-++} = p(\neg R_5 \,\&\, R_3 \,\&\, R_2),$$

$$p_{+--} = p(R_5 \,\&\, \neg R_3 \,\&\, \neg R_2),$$

$$p_{+-+} = p(R_5 \,\&\, \neg R_3 \,\&\, R_2),$$

$$p_{++-} = p(R_5 \,\&\, R_3 \,\&\, \neg R_2),$$

$$p_{+++} = p(R_5 \,\&\, R_3 \,\&\, R_2).$$

Here,

$$p_{+++} + p_{++-} + p_{+-+} + p_{+--} + p_{-++} + p_{-+-} + p_{--+} + p_{---} = 1.$$

The condition $p(R_5 \,\&\, R_2) = 0$ means that

$$p_{+++} + p_{+-+} = 0.$$

Since, all the probabilities are non-negative, the sum of two probabilities can be equal to 0 only if each of them is equal to 0, i.e.,

$$p_{+++} = p_{+-+} = 0.$$

Similarly, the condition $p(R_3 \,\&\, \neg R_2) = 0$ means that $p_{++-} + p_{-+-} = 0$. Since, the probabilities are non-negative, we conclude that

$$p_{++-} = p_{-+-} = 0.$$

We are interested in the value of

$$p(R_5 \,\&\, R_3) = p_{+++} + p_{++-}.$$

We already know that under the constraints, $p_{+++} = 0$ and $p_{++-} = 0$, hence,

$$p(R_5 \,\&\, R_3) = 0 + 0 = 0.$$

Thus, both the largest and the smallest possible value of $p(R_5 \,\&\, R_3)$ under the given constraint are 0, i.e.,

$$\mathbf{p}(R_5 \,\&\, R_3) = [0, 0].$$

3.3 Estimating the Probability Range for $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$: Second Example

Another case when we used this approach when estimating $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$ was when we estimated $p(R_5 \vee R_3)$. For that, we used two different estimates.

In the first estimate, we used the facts that $p(R_5 \,\&\, R_3) = 0$, $p(R_5) \in [0, 0.3]$, and $p(R_3) \in [0.3, 0.6]$. Here, we only have 2 variables R_5 and R_3, so we only have to consider $2^2 = 4$ possible combinations $R_5 \,\&\, R_3$, $R_5 \,\&\, \neg R_3$, $\neg R_5 \,\&\, R_3$, and $\neg R_5 \,\&\, \neg R_3$, with probabilities

$$p_{++} = p(R_5 \,\&\, R_3),$$

$$p_{+-} = (R_5 \,\&\, \neg R_3),$$

$$p_{-+} = (\neg R_5 \,\&\, R_3),$$

$$p_{--} = (\neg R_5 \,\&\, \neg R_3).$$

Here,

$$R_5 = (R_5 \,\&\, R_3) \vee (R_5 \,\&\, \neg R_3),$$

hence,

$$p(R_5) = p(R_5 \,\&\, R_3) \vee p(R_5 \,\&\, \neg R_3) = p_{++} + p_{+-}.$$

So, the fact that we know that $p(R_5) \in [0, 0.3]$ can be described as

$$0 \leq p_{++} + p_{+-} \leq 0.3.$$

Similarly,

$$R_3 = (R_5 \,\&\, R_3) \vee (\neg R_5 \,\&\, R_3)$$

hence,

$$p(R_3) = p(R_5 \,\&\, R_3) + p(\neg R_5 \,\&\, R_3) = p_{++} + p_{-+}.$$

So, the fact that we know that $p(R_3) \in [0.3, 0.6]$, means that

$$0.3 \leq p_{++} + p_{-+} \leq 0.6.$$

The information $p(R_5 \,\&\, R_3) = 0$ is directly translatable into

$$p_{++} = 0.$$

We are interested in the probability of

$$R_5 \vee R_3 = (R_5 \,\&\, R_3) \vee (R_5 \,\&\, \neg R_3) \vee (\neg R_5 \,\&\, R_3),$$

i.e., in the probability

$$p(R_5 \vee R_3) = p(R_5 \,\&\, R_3) + p(R_5 \,\&\, \neg R_3) + p(\neg R_5 \,\&\, R_3) = p_{++} + p_{+-} + p_{-+}.$$

So, to find the largest possible value of $p(R_5 \vee R_3)$, we must solve the following linear programming problem:

Maximize: $p_{++} + p_{+-} + p_{-+}$ under the constraints

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1;$$

$$0 \le p_{++} + p_{+-} \le 0.3;$$

$$0.3 \le p_{++} + p_{-+} \le 0.6;$$

$$p_{++} = 0;$$

$$p_{++} \ge 0; \quad p_{+-} \ge 0; \quad p_{-+} \ge 0; \quad p_{--} \ge 0.$$

Similarly, to find the smallest possible value of $p(R_5 \vee R_3)$, we must solve the following linear programming problem:

Minimize: $p_{++} + p_{+-} + p_{-+}$ under the constraints

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1;$$

$$0 \le p_{++} + p_{+-} \le 0.3;$$

$$0.3 \le p_{++} + p_{-+} \le 0.6;$$

$$p_{++} = 0;$$

$$p_{++} \geq 0; \quad p_{+-} \geq 0; \quad p_{-+} \geq 0; \quad p_{--} \geq 0.$$

The constraint $0 \leq p_{++} + p_{+-}$ automatically follows from $p_{++} \geq 0$; $p_{+-} \geq 0$; so it is sufficient to consider the simplified constraint

$$p_{++} + p_{+-} \leq 0.3.$$

As we have mentioned, both the smallest and the largest possible values are attained when the largest number of inequalities become equalities. In our case, we have 2 equalities:

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1$$

and

$$p_{++} = 0,$$

and 4 unknowns. Thus, we can have 2 inequalities become equalities.

In our system, we have 7 inequalities:

$$p_{++} + p_{+-} \leq 0.3;$$

$$0.3 \leq p_{++} + p_{-+};$$

$$p_{++} + p_{-+} \leq 0.6;$$

$$p_{++} \geq 0;$$

$$p_{+-} \geq 0;$$

$$p_{-+} \geq 0;$$

$$p_{--} \geq 0.$$

Since $p_{++} = 0$, the inequality

$$p_{++} \geq 0$$

is automatically satisfied, and the first three inequalities can be simplified. So, we arrive at the following 6 inequalities:

$$p_{+-} \leq 0.3;$$

$$0.3 \le p_{-+};$$

$$p_{-+} \le 0.6;$$

$$p_{+-} \ge 0;$$

$$p_{-+} \ge 0;$$

$$p_{--} \ge 0.$$

Since, $0.3 \le p_{-+}$, it automatically implies $p_{-+} \ge 0$, so we need only 5 inequalities:

$$p_{+-} \le 0.3;$$

$$0.3 \le p_{-+};$$

$$p_{-+} \le 0.6;$$

$$p_{+-} \ge 0;$$

$$p_{--} \ge 0.$$

To find the range for $p(R_5 \vee R_3)$, we must consider all possible pairs of such inequalities. Let us consider these pairs one by one.

For $p_{+-} = 0.3$ and $p_{-+} = 0.3$, we have

$$p_{--} = 1 - p_{+-} - p_{-+} - p_{++} = 1 - 0.3 - 0.3 - 0 = 0.4$$

and

$$p(R_5 \vee R_3) = p_{+-} + p_{-+} + p_{++} = 0.3 + 0.3 + 0 = 0.6.$$

For $p_{+-} = 0.3$ and $p_{-+} = 0.6$, we have

$$p_{--} = 1 - p_{+-} - p_{-+} - p_{++} = 1 - 0.3 - 0.6 - 0 = 0.1$$

and

$$p(R_5 \vee R_3) = p_{+-} + p_{-+} + p_{++} = 0.3 + 0.6 + 0 = 0.9.$$

The combination of $p_{+-} = 0.3$ and $p_{+-} = 0$ is impossible.

If $p_{+-} = 0.3$ and $p_{--} = 0$, then from $p_{++} = 0$ and

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1,$$

we conclude that

$$p_{-+} = 1 - p_{++} - p_{+-} - p_{--} = 1 - 0 - 0.3 - 0 = 0.7,$$

which contradicts to the constraint

$$p_{-+} \leq 0.6.$$

The combination of $p_{-+} = 0.3$ and $p_{-+} = 0.6$ is impossible.

If $p_{-+} = 0.3$ and $p_{+-} = 0$, then

$$p_{--} = 1 - p_{+-} - p_{-+} - p_{++} = 1 - 0 - 0.3 - 0 = 0.7$$

and

$$p(R_5 \vee R_3) = p_{+-} + p_{-+} + p_{++} = 0 + 0.3 + 0 = 0.3.$$

If $p_{-+} = 0.3$ and $p_{--} = 0$, then

$$p_{+-} = 1 - p_{-+} - p_{--} - p_{++} = 1 - 0.3 - 0 - 0 = 0.7,$$

which contradicts to our constraint $p_{+-} \leq 0.3$.

If $p_{-+} = 0.6$ and $p_{+-} = 0$, then

$$p_{--} = 1 - p_{+-} - p_{-+} - p_{++} = 1 - 0.6 - 0 - 0 = 0.4$$

and

$$p(R_5 \vee R_3) = p_{+-} + p_{-+} + p_{++} = 0 + 0.6 + 0 = 0.6.$$

If $p_{-+} = 0.6$ and $p_{--} = 0$, then

$$p_{+-} = 1 - p_{-+} - p_{--} - p_{++} = 1 - 0.6 - 0 - 0 = 0.4,$$

which contradicts to our constraint $p_{+-} \leq 0.3$.

Finally, if $p_{+-} = 0$ and $p_{--} = 0$, then

$$p_{-+} = 1 - p_{+-} - p_{--} - p_{++} = 1,$$

which contradicts to our constraint $p_{-+} \leq 0.6$.

We know that the minimum and maximum of $p(R_5 \vee R_3)$ is attained at one of the vertices. We have shown that at the vertices, $p(R_5 \vee R_3)$ takes the values 0.6, 0.9, 0.3, and 0.6. The smallest of these values is 0.3 and the largest of these values is 0.9. So, we conclude that the range of possible values of $p(R_5 \vee R_3)$ is contained in the interval $[0.3, 0.9]$.

3.4 Estimating the Probability Range for $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$: Third Example

In the estimation presented in Chapter 2 when we estimated the probability $p(R_5 \vee, R_3)$, we used the formulas $\mathbf{p}(R_3 \,\&\, \neg R_1) = [0, 0]$, $\mathbf{p}(R_5 \,\&\, \neg R_1) = [0, 0]$, $\mathbf{p}(R_1) = [0.6, 0.6]$, and $\mathbf{p}(R_3) = [0.3, 0.6]$, to conclude that $\mathbf{p}(R_5 \vee R_3) = [0.3, 0.6]$.

Here, we have 3 basic statements used: R_5, R_3, and R_1, so we have $2^3 = 8$ possible combinations

$$R_5^{\varepsilon_5} \,\&\, R_3^{\varepsilon_3} \,\&\, R_1^{\varepsilon_1},$$

and thus, 8 unknowns:

$$p_{---} = p(\neg R_5 \,\&\, \neg R_3 \,\&\, \neg R_1),$$

$$p_{--+} = p(\neg R_5 \,\&\, \neg R_3 \,\&\, R_1),$$

$$p_{-+-} = p(\neg R_5 \,\&\, R_3 \,\&\, \neg R_1),$$

$$p_{-++} = p(\neg R_5 \,\&\, R_3 \,\&\, R_1),$$

$$p_{+--} = p(R_5 \,\&\, \neg R_3 \,\&\, \neg R_1),$$

$$p_{+-+} = p(R_5 \,\&\, \neg R_3 \,\&\, R_1),$$

$$p_{++-} = p(R_5 \,\&\, R_3 \,\&\, \neg R_1),$$

$$p_{+++} = p(R_5 \,\&\, R_3 \,\&\, R_1).$$

Here,

$$p_{+++} + p_{++-} + p_{+-+} + p_{+--} + p_{-++} + p_{-+-} + p_{--+} + p_{---} = 1.$$

The condition $p(R_5 \,\&\, \neg R_1) = 0$ means that

$$p_{++-} + p_{+--} = 0.$$

Since all the probabilities are non-negative, the sum of two probabilities can be equal to 0 only if each of them is equal to 0, i.e.,

$$p_{++-} = p_{+--} = 0.$$

Similarly, the condition $p(R_3 \,\&\, \neg R_1) = 0$ means that $p_{++-} + p_{-+-} = 0$. Since, the probabilities are non-negative, we conclude that

$$p_{++-} = p_{-+-} = 0.$$

Here,

$$R_3 = (R_5 \,\&\, R_3 \,\&\, R_1) \vee (\neg R_5 \,\&\, R_3 \,\&\, R_1) \vee (R_5 \,\&\, R_3 \,\&\, \neg R_1) \vee (\neg R_5 \,\&\, R_3 \,\&\, \neg R_1),$$

hence,

$$p(R_5) = p(R_5 \,\&\, R_3 \,\&\, R_1) + p(\neg R_5 \,\&\, R_3 \,\&\, R_1) +$$

$$p(R_5 \,\&\, R_3 \,\&\, \neg R_1) + p(\neg R_5 \,\&\, R_3 \,\&\, \neg R_1) = p_{+++} + p_{-++} + p_{++-} + p_{-+-}.$$

The condition $\mathbf{p}(R_3) = [0.3, 0.6]$ can be therefore described as follows:

$$0.3 \le p_{+++} + p_{-++} + p_{++-} + p_{-+-} \le 0.6.$$

The condition that $\mathbf{p}(R_1) = [0.6, 0.6]$ can be described as

$$p_{+++} + p_{--+} + p_{+-+} + p_{-++} = 0.6.$$

We are interested in the probability of

$$R_5 \vee R_3 = (R_5 \,\&\, R_3 \,\&\, R_1) \vee (R_5 \,\&\, \neg R_3 \,\&\, R_1) \vee (R_5 \,\&\, R_3 \,\&\, \neg R_1) \vee$$

$$(R_5 \,\&\, \neg R_3 \,\&\, \neg R_1) \vee (\neg R_5 \,\&\, R_3 \,\&\, \neg R_1) \vee (\neg R_5 \,\&\, R_3 \,\&\, \neg R_1),$$

i.e., in the probability

$$p(R_5 \vee R_3) = p(R_5 \,\&\, R_3 \,\&\, R_1) + p(R_5 \,\&\, \neg R_3 \,\&\, R_1) + p(R_5 \,\&\, R_3 \,\&\, \neg R_1) +$$

$$p(R_5 \,\&\, \neg R_3 \,\&\, \neg R_1) + p(\neg R_5 \,\&\, R_3 \,\&\, \neg R_1) + p(\neg R_5 \,\&\, R_3 \,\&\, \neg R_1) =$$

$$p_{+++} + p_{+-+} + p_{++-} + p_{+--} + p_{-++} + p_{-+-}.$$

So, to find the largest possible value of $p(R_5 \vee R_3)$, we must solve the following linear programming problem:

Maximize:

$$p_{+++} + p_{+-+} + p_{++-} + p_{+--} + p_{-++} + p_{-+-}$$

under the constraints

$$p_{+++} + p_{++-} + p_{+-+} + p_{+--} + p_{-++} + p_{-+-} + p_{--+} + p_{---} = 1;$$

$$p_{+++} + p_{--+} + p_{+-+} + p_{-++} = 0.6;$$

$$0.3 \le p_{+++} + p_{-++} + p_{++-} + p_{-+-} \le 0.6;$$

$$p_{++-} = p_{+--} = 0;$$

$$p_{++-} = p_{-+-} = 0;$$

$$p_{+++} \ge 0; \quad p_{+-+} \ge 0; \quad p_{-++} \ge 0; \quad p_{--+} \ge 0; \quad p_{---} \ge 0.$$

Similarly, to find the smallest possible value of $p(R_5 \vee R_3)$, we must solve the following linear programming problem:

Minimize:

$$p_{+++} + p_{+-+} + p_{++-} + p_{+--} + p_{-++} + p_{-+-}$$

under the constraints

$$p_{+++} + p_{++-} + p_{+-+} + p_{+--} + p_{-++} + p_{-+-} + p_{--+} + p_{---} = 1;$$

$$p_{+++} + p_{--+} + p_{+-+} + p_{-++} = 0.6;$$

$$0.3 \leq p_{+++} + p_{-++} + p_{++-} + p_{-+-} \leq 0.6;$$

$$p_{++-} = p_{+--} = 0;$$

$$p_{-+-} = 0;$$

$$p_{+++} \geq 0; \quad p_{+-+} \geq 0; \quad p_{-++} \geq 0; \quad p_{--+} \geq 0; \quad p_{---} \geq 0.$$

Since $p_{++-} = p_{+--} = p_{-+-} = 0$, we have 5 unknowns, and the problems take the following form:

Maximize:

$$p_{+++} + p_{+-+} + p_{-++}$$

under the constraints

$$p_{+++} + p_{+-+} + p_{-++} + p_{--+} + p_{---} = 1;$$

$$p_{+++} + p_{--+} + p_{+-+} + p_{-++} = 0.6;$$

$$0.3 \leq p_{+++} + p_{-++} \leq 0.6;$$

$$p_{+++} \geq 0; \quad p_{+-+} \geq 0; \quad p_{-++} \geq 0; \quad p_{--+} \geq 0; \quad p_{---} \geq 0.$$

and

Minimize:

$$p_{+++} + p_{+-+} + p_{-++}$$

under the constraints

$$p_{+++} + p_{+-+} + p_{-++} + p_{--+} + p_{---} = 1;$$

$$p_{+++} + p_{--+} + p_{+-+} + p_{-++} = 0.6;$$

$$0.3 \le p_{+++} + p_{-++} \le 0.6;$$

$$p_{+++} \ge 0; \quad p_{+-+} \ge 0; \quad p_{-++} \ge 0; \quad p_{--+} \ge 0; \quad p_{---} \ge 0.$$

These LP problems can be further simplified if we notice that the difference between

$$p_{+++} + p_{+-+} + p_{-++} + p_{--+} + p_{---}$$

and

$$p_{+++} + p_{--+} + p_{+-+} + p_{-++}$$

is exactly p_{---}, so we can replace the first equality with $p_{---} = 0.4$.

The objective function does not depend on p_{---}, and neither of the constraints depend on p_{---}, so we can simply ignore the variable p_{---} and reformulate our problems in terms of the remaining 4 variables:

Maximize:

$$p_{+++} + p_{+-+} + p_{-++}$$

under the constraints

$$p_{+++} + p_{+-+} + p_{-++} + p_{--+} = 0.6;$$

$$0.3 \le p_{+++} + p_{-++} \le 0.6;$$

$$p_{+++} \ge 0; \quad p_{+-+} \ge 0; \quad p_{-++} \ge 0; \quad p_{--+} \ge 0.$$

and

Minimize:

$$p_{+++} + p_{+-+} + p_{-++}$$

under the constraints

$$p_{+++} + p_{+-+} + p_{-++} + p_{--+} = 0.6;$$

$$0.3 \le p_{+++} + p_{-++} \le 0.6;$$

$$p_{+++} \ge 0; \quad p_{+-+} \ge 0; \quad p_{-++} \ge 0; \quad p_{--+} \ge 0.$$

In our constraints, we have linear equality and 4 unknowns. In general, once we have as many equalities as unknowns, we can get a unique solution. So for 4 unknowns, we can have at most 4 linear equalities. Thus, we can have at most 3 inequalities becoming equalities.

So, to find the minimum and the maximum of $p(R_5 \vee R_3)$, we must consider all possible triples of inequalities.

Out of two inequalities

$$0.3 \le p_{+++} + p_{-++}$$

and

$$p_{+++} + p_{-++} \le 0.6,$$

only one can be an equality. So, we must consider two types of cases:

- case when all 3 equalities are of the form $p_{\pm\pm\pm} = 0$;
- case when one of the equalities is of the type $p_{+++}+p_{-++} = 0.3$ or $p_{+++}+p_{-++} = 0.6$.

Let us consider these cases one by one.

If 3 out of 4 variables are 0, this means that only one is allowed to be non-0. This one has to be either p_{+++} or p_{-++}, because otherwise the sum $p_{+++} + p_{-++}$ would be equal to 0, and we know that

$$0.3 \le p_{+++} + p_{-++}.$$

So, in the situations of the first type, we only need to consider two possibilities:

- $p_{+++} \neq 0$ and 3 other variables are 0;
- $p_{-++} \neq 0$ and 3 other variables are 0.

In the first case, the original equality leads to

$$p_{+++} = 0.6.$$

Here, $p_{+++} + p_{-++} = 0 + 0.6 = 0.6$, so the inequality

$$0.3 \le p_{+++} + p_{-++} \le 0.6$$

is satisfied. In this case, the objective function $p_{+++} + p_{+-+} + p_{-++}$ attains the value $0.6 + 0 + 0 = 0.6$.

In the second case, the original equality leads to

$$p_{-++} = 0.6.$$

Here, $p_{+++} + p_{-++} = 0.6 + 0 = 0.6$, so the inequality

$$0.3 \le p_{+++} + p_{-++} \le 0.6$$

is satisfied. In this case, the objective function $p_{+++} + p_{+-+} + p_{-++}$ attains the value $0 + 0 + 0.6 = 0.6$.

Let us now consider the cases when $p_{+++} + p_{-++} = 0.3$. Subtracting this equality from the original one, we can reformulate the original equality in the following simplified form:

$$p_{+-+} + p_{--+} = 0.6 - 0.3 = 0.3.$$

We need two inequalities to become equalities, so we need two more equalities of the type $p_{\pm\pm\pm} = 0$. We cannot have $p_{+++} = p_{-++} = 0$, because otherwise, we would get $p_{+++} = p_{-++} = 0 + 0 = 0 \ne 0.3$.

So, we must pick one $p_{\pm\pm\pm}$ from each of these pairs to be equal to 0. Let us describe all $2 \times 2 = 4$ possible picks.

If $p_{+++} = 0$ and $p_{+-+} = 0$, then

$$p_{-++} = p_{--+} = 0.3 - 0 = 0.3,$$

and

$$p_{+++} + p_{+-+} + p_{-++} = 0.3.$$

If $p_{+++} = 0$ and $p_{--+} = 0$, then

$$p_{-++} = p_{+-+} = 0.3,$$

and

$$p_{+++} + p_{+-+} + p_{-++} = 0.6.$$

If $p_{-++} = 0$ and $p_{+-+} = 0$, then

$$p_{+++} = p_{--+} = 0.3,$$

so

$$p_{+++} + p_{+-+} + p_{-++} = 0.3 + 0 + 0 = 0.3.$$

Finally, if $p_{-++} = 0$ and $p_{--+} = 0$, then

$$p_{+++} = p_{+-+} = 0.3,$$

so

$$p_{+++} + p_{+-+} + p_{-++} = 0.3 + 0.3 + 0 = 0.6.$$

To complete this case, we must consider situations when $p_{+++} + p_{-++} = 0.6$.

Subtracting this equality from the original one, we can conclude that

$$p_{+-+} + p_{--+} = 0.6 - 0.6 = 0,$$

so

$$p_{+-+} = p_{--+} = 0.$$

In this case, the objective function takes the form

$$p_{+++} + p_{+-+} + p_{-++} = p_{+++} + p_{-++} = 0.6.$$

We know that the minimum and maximum of $p(R_5 \vee R_3)$ is attained at one of the vertices. We have shown that at the vertices, $p(R_5 \vee R_3)$ takes the values 0.3 or 0.6. The smallest of these values is 0.3 and the largest of these values is 0.6. So, we conclude that the range of possible values of $p(R_5 \vee R_3)$ is contained in the interval $[0.3, 0.6]$.

Chapter 4

Software Implementation of the New Algorithm

4.1 Linear Programming Tool - lpsolve

Linear programming (LP) is a known problem for which many tools exist. To find the best tool, I checked on Google and got an impression that out of free tools, one was most widely used – `lpsolve` (also known as LPSOLVER). `lpsolve` is written in ANSI C; it can be called as a library not only from C, but from Java, Excel, Virtual Basic, and many other languages.

`lpsolve` can also solve integer programming problems, but we will only use its ability to solve linear programming problems.

This program was developed by Michel Berkelaar, Kjell Eikland, Peter Notebaert under the GNU LGPL (Lesser General Public License). In this thesis, I am using version 5.0.10.1 dated 1 May 2004.

`lpsolve` can be downloaded from a web-group on `www.yahoo.com`. This is an free open source software, but you do need to register at `www.yahoo.com` to get access to this information. Once you have a `yahoo.com` login, you can login with your username and password at `http://groups.yahoo.com/group/lp_solve/`. Once logged in, you reach the home page of the `lpsolve` group.

On this home page is a description of the software and an explanation of the different links on the website. As with many free software packages, the instructions on the website are not very user-friendly. Let us therefore describe here how to download this package.

The files that you need to download to run `lpsolve` are under the `Files` link. Under the `Files` link, you need to click on version 5.0 and look for the `lpsolve` source code file–`lp_solve_5.0_source.tar.gz`.

Next, download this file to the hard drive of your computer and unzip the file into a folder. Another file that needs to be downloaded from the same link is the `LpSolveIDE.zip` which needs to be downloaded and extracted to the same folder on your computer's hard drive where the source code is saved.

Next, reboot the computer and click on the `LpSolveIDE.exe` file and it will open up the Integrated Development Environment (IDE) to access the `lpsolve`.

4.2 Input File Format Needed for `lpsolve`

The `lpsolve` Integrated Development Environment (IDE) needs the input files to be in a particular input format called `lp-format`. A file in an `lp-format` consists of the following 3 parts:

- objective function;
- constraints;
- variable declaration and bounds (optional);

An objective function forms the first line of the file; it is a linear combination of variables and constants ending by a semicolon, e.g.:

$$\mathtt{x1} + \mathtt{x2} + 3;$$

or

$$2 \cdot \mathtt{x1} + \mathtt{y} - \mathtt{z};$$

By default, the program maximizes the objective function. If we want to minimize, we should write `min:` (or, alteratively, `Min:`, `MIN:`, `minimize:`, or `Minimise:`) before the actual expression for the objective function; e.g.:

$$\texttt{Min} : \texttt{x1} + \texttt{x2} + 3;$$

For clarification, `lpsolve` also allows the user to explicitly explain that we want maximization, by writing `max:`, `Max:`, `MAX:`, `maximize:`, or `Maximise:` before the actual expression for the objective function; e.g.:

$$\texttt{Max} : \texttt{x1} + \texttt{x2} + 3;$$

Each constraint is contained in a separate line; it contains a linear combination followed by a relation symbol and a constant, e.g.:

$$\texttt{x1} + 3 * \texttt{x2} \leq 0.6;$$

Instead of $\leq$ or $\geq$, the system allows the user to simply write $<$ or $>$. So, in the above example, we can describe the same inequality as

$$\texttt{x1} + 3 * \texttt{x2} < 0.6;$$

If one of the constraints is an equality, then we describe it as

$$\texttt{x1} + 3 * \texttt{x2} = 0.6;$$

Alteratively, we can describe the same constraint as

$$0.6 \geq \texttt{x1} + 3 * \texttt{x2};$$

If for some linear expression, we have bounds on both sides, then we can represent both bounds in a single constraint; e.g.,

$$0 \leq \texttt{x1} + 3 * \texttt{x2} \leq 0.6;$$

By default, the system treats all variable names in the objective function and in the constraints as real numbers. If we want to indicate that some of these variables only take integer values, we can do that by explicitly describing it in the optional declaration part of an lp-file, e.g.,

```
int x1, x2;
```

It is also possible to include variable bounds in the declarations and bounds part; this additional feature is added to `lpsolve` to make it computationally efficient in solving large-size LP problems. Since our LP problems are relatively small, we will not use this feature.

The input file can be prepared beforehand or explicitly typed in line by line into the input window of the `lpsolve` tool. When you type in, the header

```
/* Objective function */
```

and

```
/* Variable bounds */
```

automatically appear.

4.3 How to Use `lpsolve`: Instruction and Example

To use `lpsolve`, we need to input the problem into the `lpsolve` tool. This can be done either by inputting a ready file or by typing in the problem into the input window.

Once the problem is entered, we go to the `Action` menu and click on `Solve`. The `Log` window describes what the computer did and how long it took. The solution itself is viewed by clicking on the `Result` button.

Let us illustrate the use of `lpsolve` on a toy example. Let us assume that we want to minimize the objective function

$$x_1 + x_2 + 3$$

under the constraints

$$x_1 \geq 1;$$

$$x_2 \geq 1.$$

Let us also assume that both variables x_1 and x_2 are integers.

In this example, the smallest possible value of $x_1 + x_2 + 3$ is attained when both x_1 and x_2 take their smallest possible values $x_1 = 1$ and $x_2 = 1$. The resulting minimal value of the objective function is

$$x_1 + x_2 + 3 = 1 + 1 + 3 = 5.$$

In accordance with the above instructions, this problem is represented as the following lp-file:

```
/* Objective function */
Min: x_1 + x_2 + 3;
/* Constraints */
x_1 > 1;
x_2 > 1;
/* Variable bounds */
int x_1, x_2;
```

It is worth mentioning that here, $>$ actually means $\geq$.

Once we click on `Solve`, the following log appears; see Figure 4.1. In the process of solving the given LP problem, the LP solver transforms the original problem into a matrix form:

$$a_1 \cdot x_1 + a_2 \cdot x_2 + \ldots + a_n \cdot x_n + a_0 \to \max \quad (\min)$$

under the constraints of the type

$$c_{11} \cdot x_1 + c_{12} \cdot x_2 + \ldots + c_{1n} \cdot x_n \geq c_{10};$$

$$c_{21} \cdot x_1 + c_{21} \cdot x_2 + \ldots + c_{2n} \cdot x_n \geq c_{20};$$

$$\ldots$$

$$c_{m1} \cdot x_1 + c_{m2} \cdot x_2 + \ldots + c_{mn} \cdot x_n \geq c_{m0}.$$

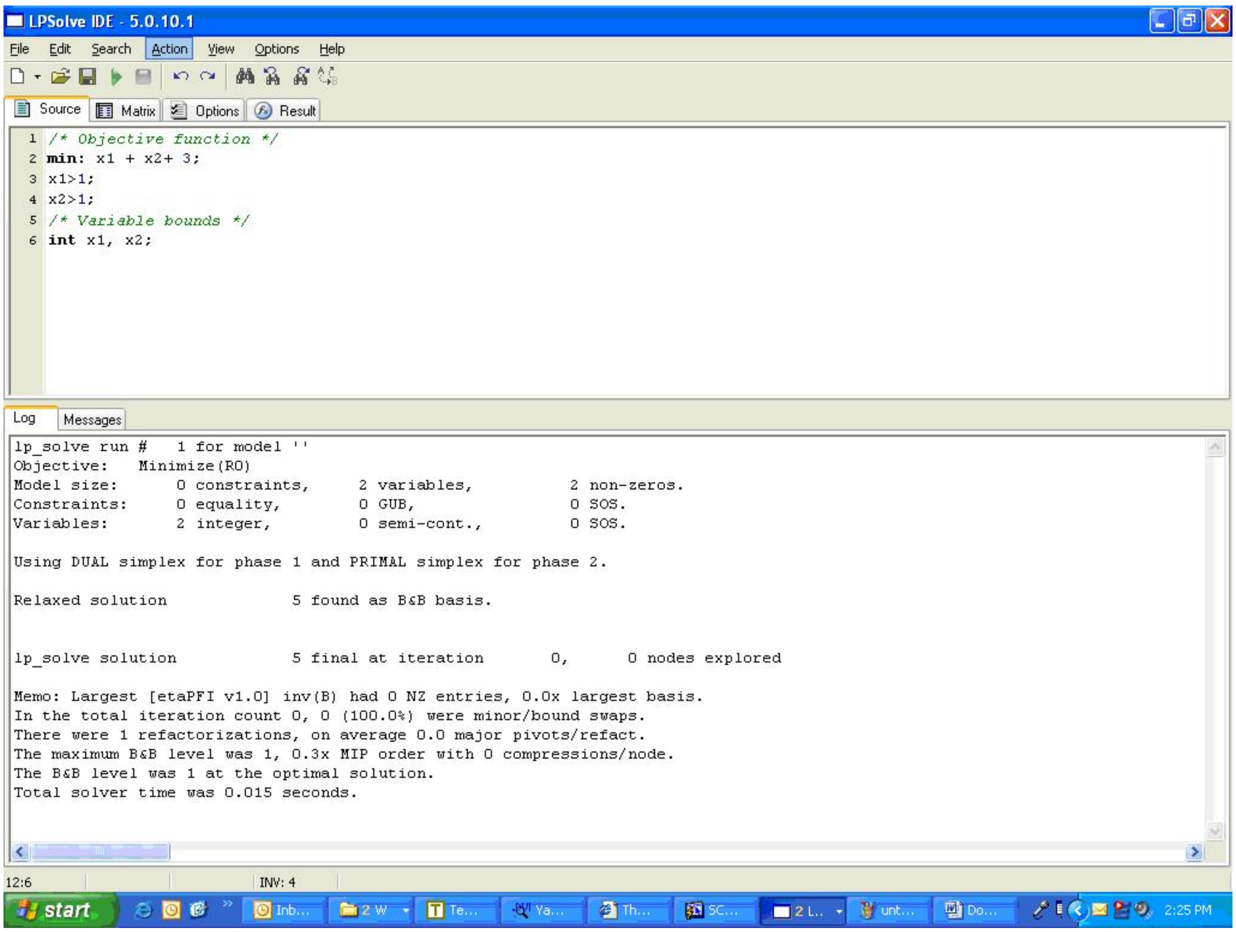

Figure 4.1: Toy Example: Input and Log

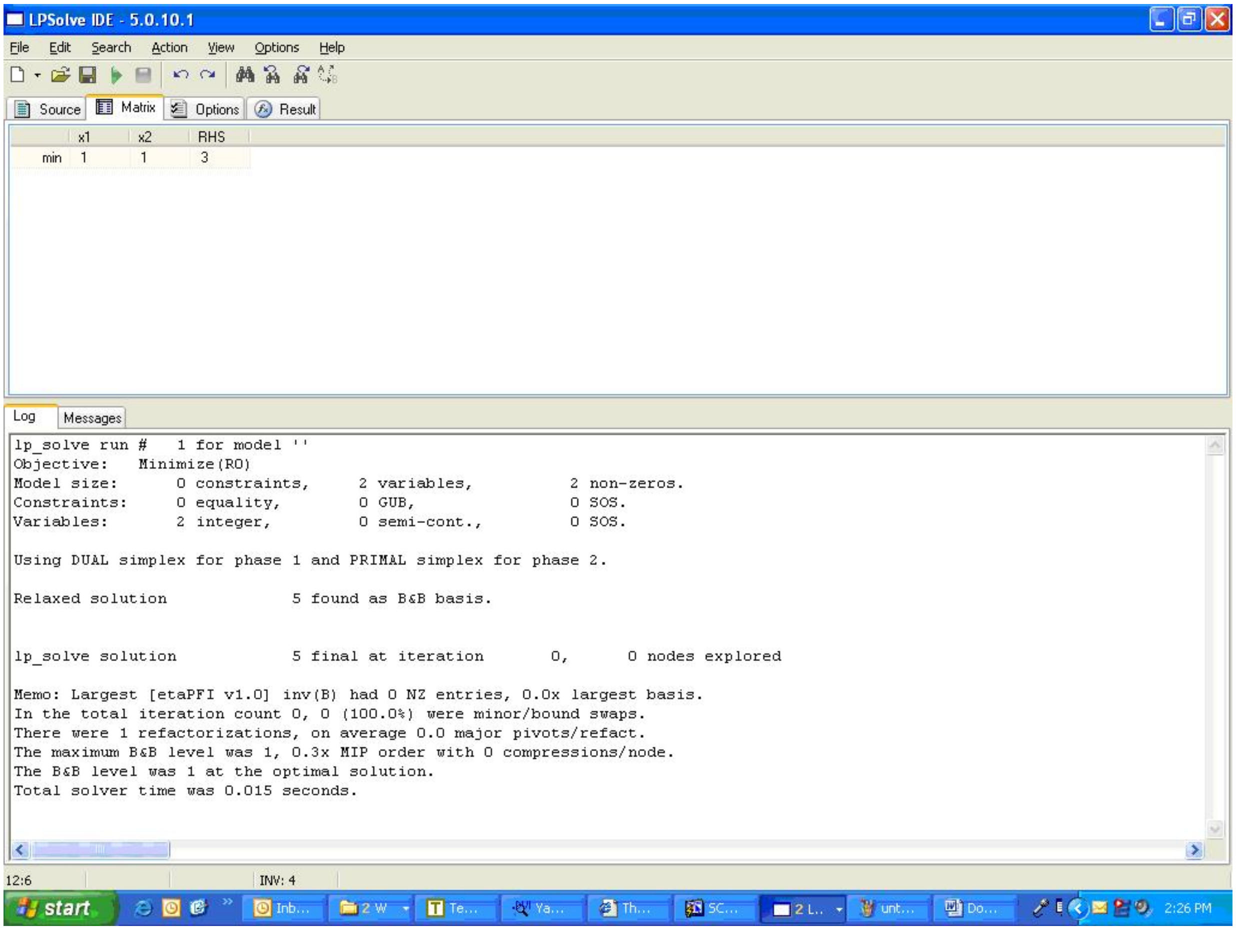

Figure 4.2: Toy Example: Matrix Window

The tool presents the coefficients a_i and c_{ij} of this representations in a separate `Matrix` window; constraints on individual variables, like $x_1 \geq 1$, are ignored.

For our toy problem, both constraints are thus ignored, and the matrix window only contains the coefficients of the objective function; see Figure 4.2. The results appears in the `Result` window, see Figure 4.3.

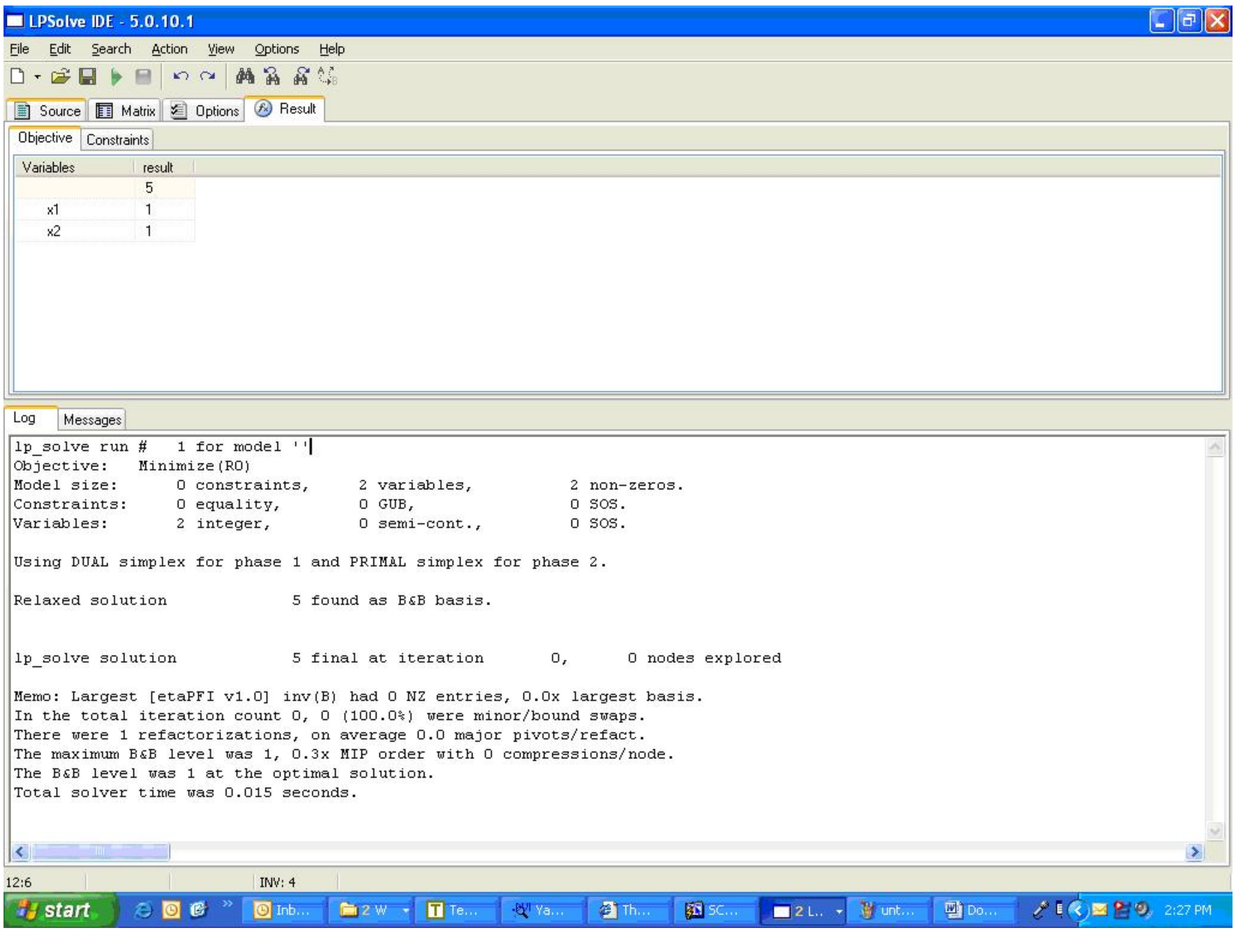

Figure 4.3: Toy Example: Result Window

4.4 Example: Estimating the Range of $(A \vee B)$

Let us start with the following simple example. We know that $p(A) = a = 0.6$ and $p(B) = b = 0.6$, and we want to find the range of $(A \vee B)$.

In this case, as we have mentioned in Chapter 3, we arrive at the following linear programming problems:

Maximize

$$p_{++} + p_{+-} + p_{-+}$$

under the constraints

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1;$$

$$p_{++} + p_{+-} = a;$$

$$p_{++} + p_{-+} = b;$$

$$p_{++} \geq 0; \quad p_{+-} \geq 0; \quad p_{-+} \geq 0; \quad p_{--} \geq 0;$$

and

Minimize

$$p_{++} + p_{+-} + p_{-+}$$

under the constraints

$$p_{++} + p_{+-} + p_{-+} + p_{--} = 1;$$

$$p_{++} + p_{+-} = a;$$

$$p_{++} + p_{-+} = b;$$

$$p_{++} \geq 0; \quad p_{+-} \geq 0; \quad p_{-+} \geq 0; \quad p_{--} \geq 0.$$

Let us denote $Ppp = p_{++}$, $Ppm = p_{+-}$, $Pmp = p_{-+}$, and $Pmm = p_{--}$. In these notations, the above linear programming problems are represented by the following lp-files:

```
/* Objective function */
max: Ppp + Ppm + Pmp;
```

```
/* Constraints */
Ppp+ Ppm + Pmp + Pmm = 1;
Ppp + Ppm = 0.6;
Ppp + Pmp < 0.6;
Ppp > 0;
Ppm > 0;
Pmp > 0;
Pmm > 0;
/* Variable bounds */
```

and

```
/* Objective function */
min: Ppp + Ppm + Pmp;
/* Constraints */
Ppp+ Ppm + Pmp + Pmm = 1;
Ppp + Ppm = 0.6;
Ppp + Pmp < 0.6;
Ppp > 0;
Ppm > 0;
Pmp > 0;
Pmm > 0;
/* Variable bounds */
```

For the first problem, `lpsolve` returns 1.0; for the second problem, `lpsolve` returns 0.6 – exactly as we described in Chapter 3.

For the minimization problem, the Input/log window, the matrix window containing the coefficients of the objective function, and the result window are shown in Figure 4.4, 4.5, and 4.6.

For the maximization problem, the Input/log window, the matrix window containing

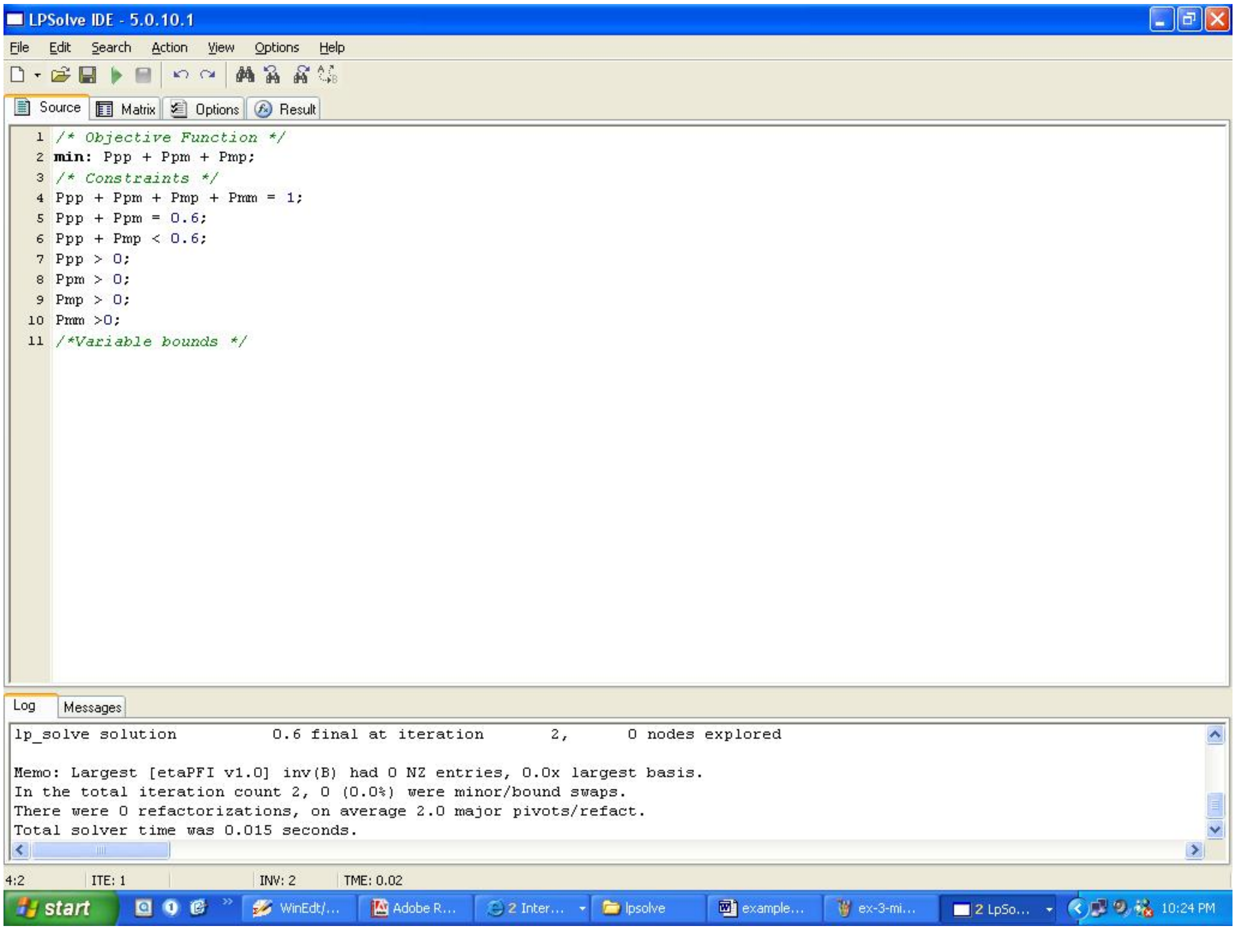

Figure 4.4: $A \vee B$: Minimization - Input/log Window

the coefficients of the objective function, and the result window are shown in Figure 4.7, 4.8, and 4.9.

4.5 Example: Estimating the Range of $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$

In Chapter 3, for the estimation to find the range for $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$, we estimated the probability $p(R_5 \vee, R_3)$, and we used the formulas $\mathbf{p}(R_3 \,\&\, \neg R_1) = [0, 0]$, $\mathbf{p}(R_5 \,\&\, \neg R_1) =$

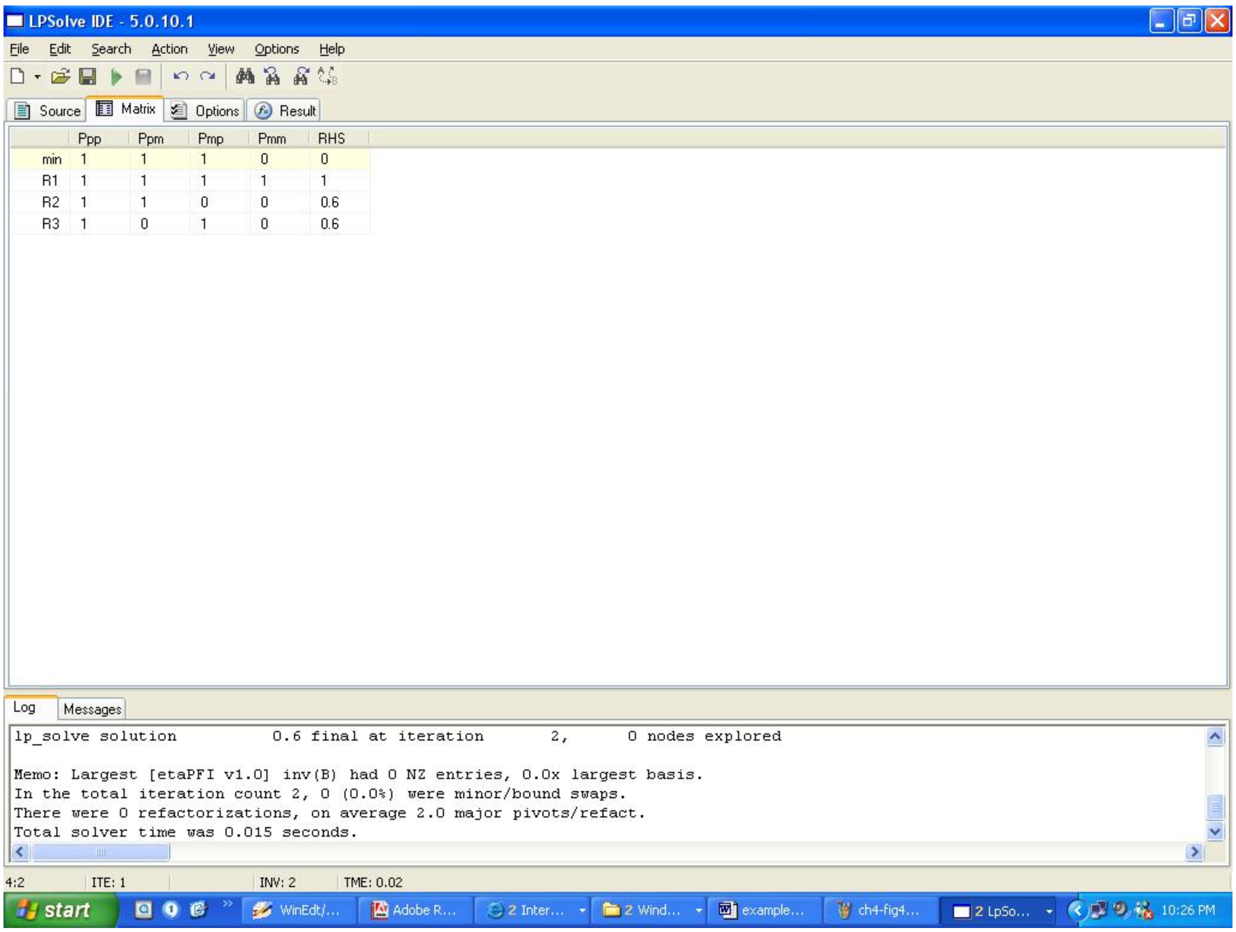

Figure 4.5: $A \vee B$: Minimization - Matrix Window

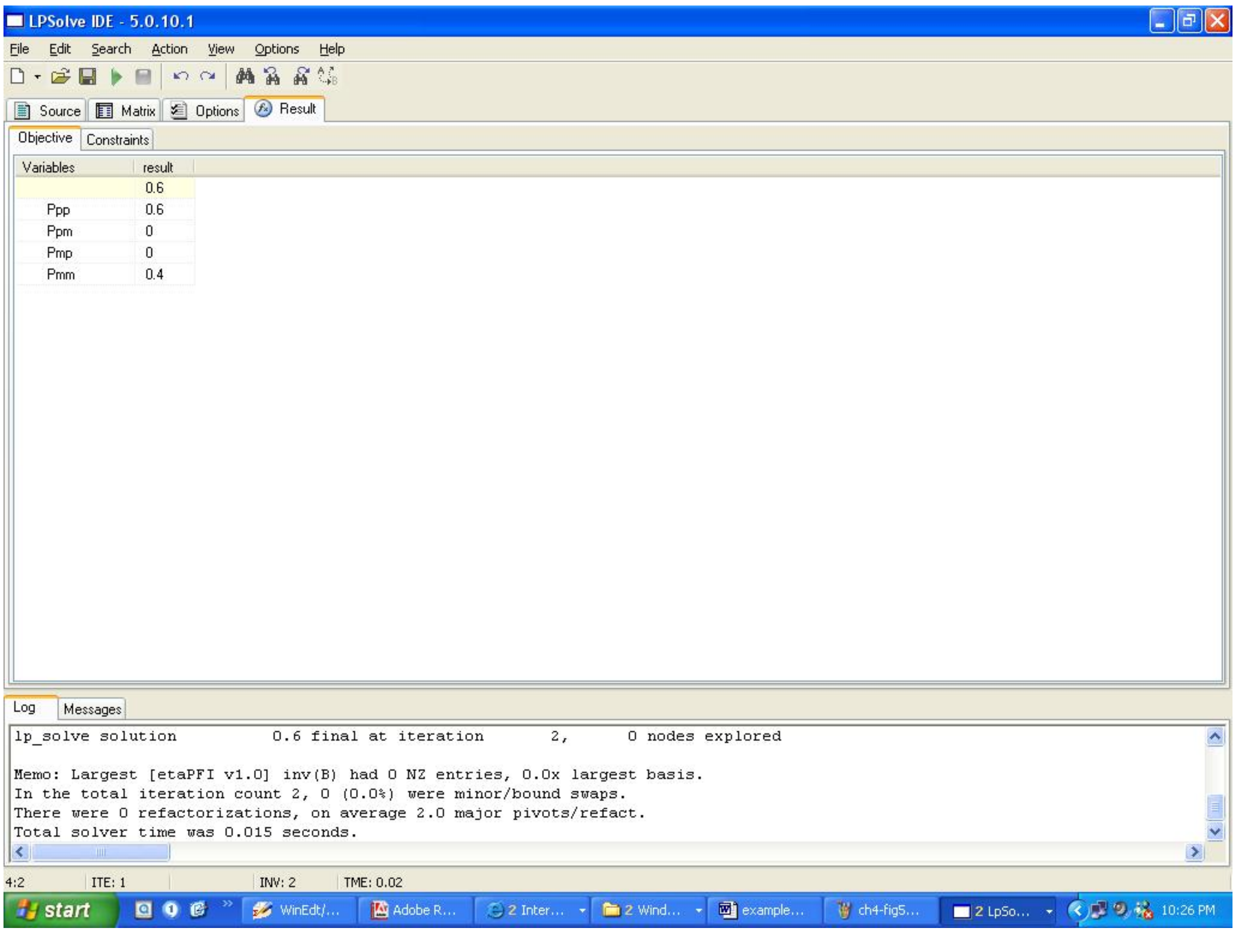

Figure 4.6: $A \vee B$: Minimization - Result Window

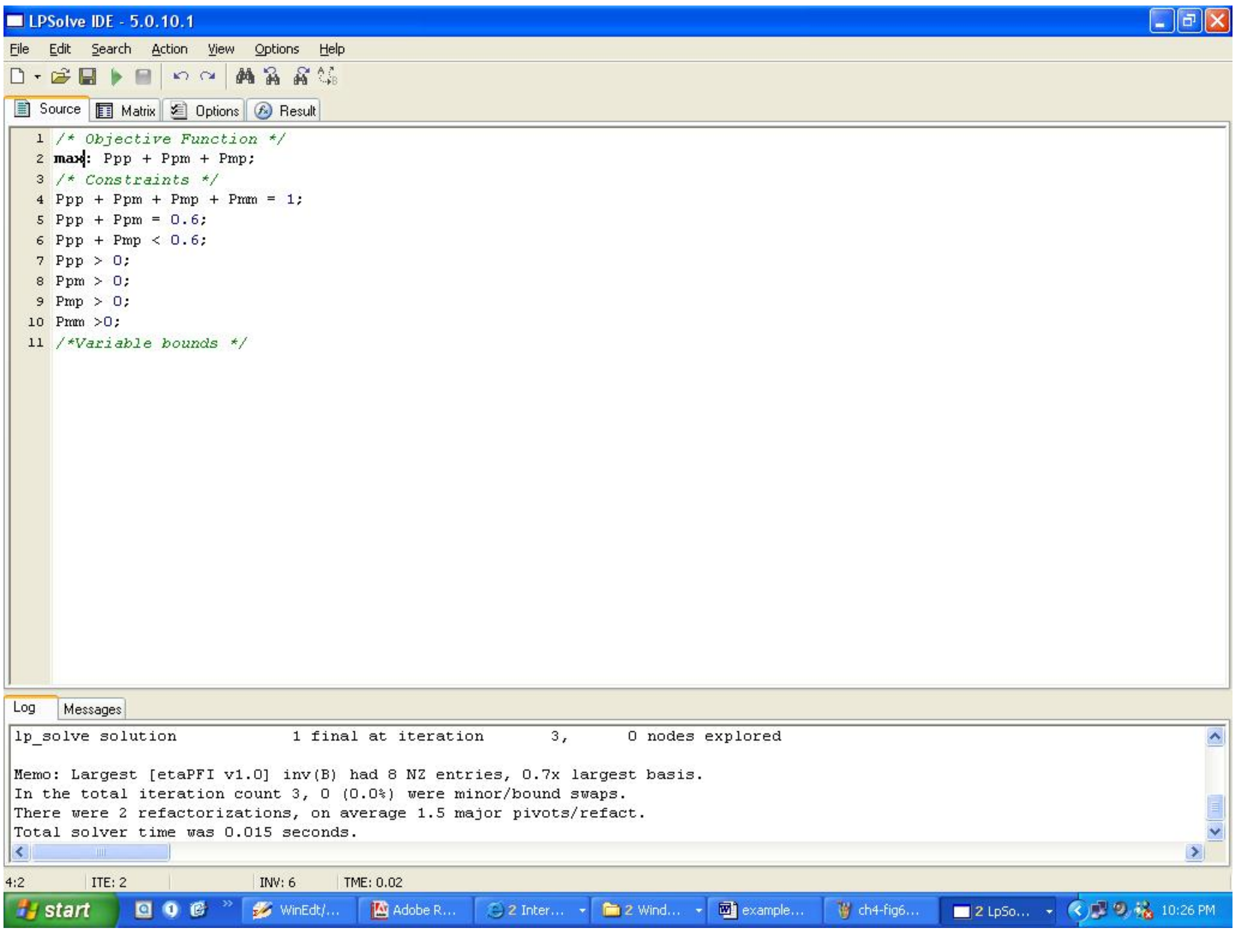

Figure 4.7: $A \vee B$: Maximization - Input/log Window

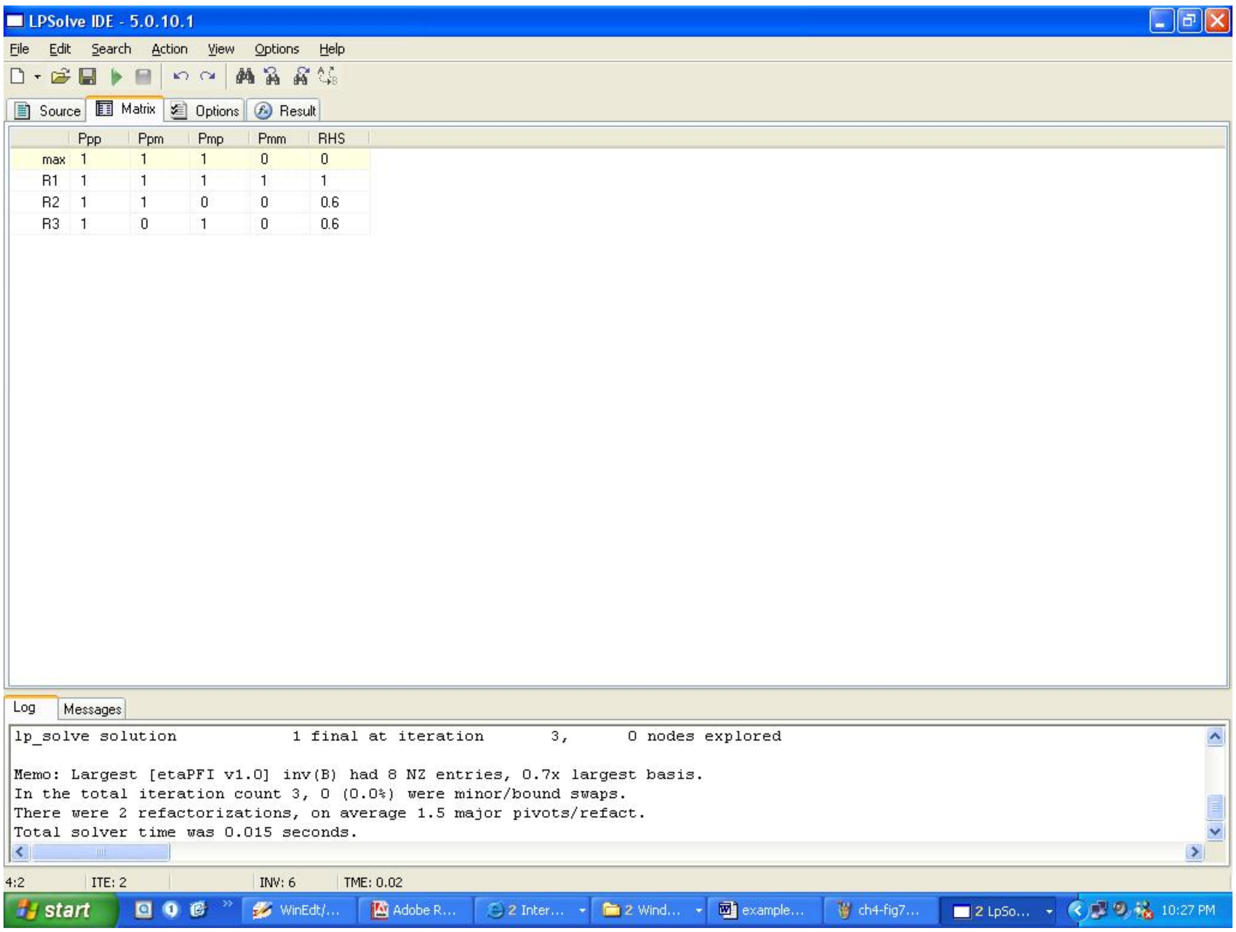

Figure 4.8: $A \vee B$: Maximization: Matrix Window

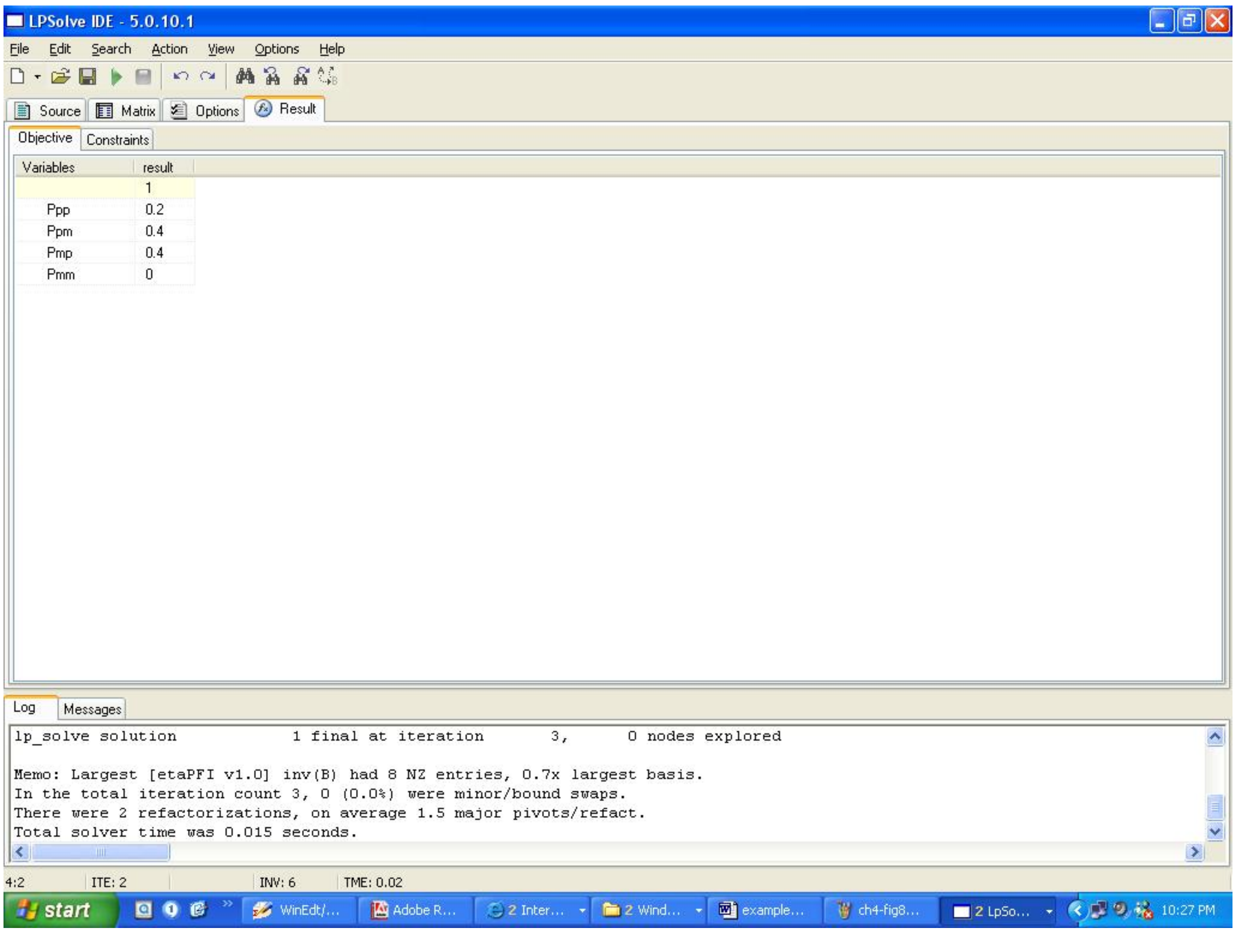

Figure 4.9: $A \vee B$: Maximization: Matrix Window

$[0,0]$, $\mathbf{p}(R_3) = [0.3, 0.6]$, and $\mathbf{p}(R_1) = [0.6, 0.6]$ to conclude that $\mathbf{p}(R_5 \vee R_3) = [0.3, 0.6]$.

In this case, we arrived at the following linear programming problems:

Maximize:

$$p_{+++} + p_{+-+} + p_{++-} + p_{+--} + p_{-++} + p_{-+-}$$

under the constraints

$$p_{+++} + p_{++-} + p_{+-+} + p_{+--} + p_{-++} + p_{-+-} + p_{--+} + p_{---} = 1;$$

$$p_{+++} + p_{--+} + p_{+-+} + p_{-++} = 0.6;$$

$$0.3 \le p_{+++} + p_{-++} + p_{++-} + p_{-+-} \le 0.6;$$

$$p_{++-} = p_{+--} = 0;$$

$$p_{-+-} = 0;$$

$$p_{+++} \ge 0; \quad p_{+-+} \ge 0; \quad p_{-++} \ge 0; \quad p_{--+} \ge 0; \quad p_{---} \ge 0;$$

and

Minimize:

$$p_{+++} + p_{+-+} + p_{++-} + p_{+--} + p_{-++} + p_{-+-}$$

under the constraints

$$p_{+++} + p_{++-} + p_{+-+} + p_{+--} + p_{-++} + p_{-+-} + p_{--+} + p_{---} = 1;$$

$$p_{+++} + p_{--+} + p_{+-+} + p_{-++} = 0.6;$$

$$0.3 \le p_{+++} + p_{-++} + p_{++-} + p_{-+-} \le 0.6;$$

$$p_{++-} = p_{+--} = 0;$$

$$p_{-+-} = 0;$$

$$p_{+++} \ge 0; \quad p_{+-+} \ge 0; \quad p_{-++} \ge 0; \quad p_{--+} \ge 0; \quad p_{---} \ge 0.$$

Let us denote $Pppp = p_{+++}$, $Pppm = p_{++-}$, $Ppmp = p_{+-+}$, $Ppmm = p_{+--}$, $Pmpp = p_{-++}$, $Pmpm = p_{-+-}$, $Pmmp = p_{--+}$, and $Pmmm = p_{+-+}$,. In these notations, the above linear programming problems are represented by the following lp-files:

```
/* Objective function */
min: Pppp + Ppmp + Pppm + Ppmm + Pmpp + Pmpm;
Pppp + Pppm + Ppmp + Ppmm + Pmpp + Pmpm + Pmmp + Pmmm = 1;
Pppp + Pmmp + Ppmp + Pmpp = 0.6;
0.3 < Pppp + Pmpp + Pppm + Pmpm < 0.6;
0 < Pppp + Ppmp + Pppm + Ppmm < 0.3;
Pppm = 0;
Ppmm = 0;
Pmpm = 0;
Pppp > 0;
Ppmp > 0;
Pmpp > 0;
Pmmp > 0;
Pmmm > 0;
/* Variable bounds */
```

and

```
/* Objective function */
max: Pppp + Ppmp + Pppm + Ppmm + Pmpp + Pmpm;
Pppp + Pppm + Ppmp + Ppmm + Pmpp + Pmpm + Pmmp + Pmmm = 1;
Pppp + Pmmp + Ppmp + Pmpp = 0.6;
0.3 < Pppp + Pmpp + Pppm + Pmpm < 0.6;
0 < Pppp + Ppmp + Pppm + Ppmm < 0.3;
Pppm = 0;
Ppmm = 0;
Pmpm = 0;
Pppp > 0;
Ppmp > 0;
```

```
Pmpp > 0;
Pmmp > 0;
Pmmm > 0;
/* Variable bounds */
```

For the minimization problem, `lpsolve` returns 0.3; for the maximization problem `lpsolve` returns 0.6 – exactly as we described in Chapter 3.

For the minimization problem, the Input/log window, the matrix window containing the coefficients of the objective function and the result window are shown in Figure 4.10, 4.11, and 4.12.

For the maximization problem, the Input/log window, the matrix window containing the coefficients of the objective function and the result window are shown in Figure 4.13, 4.14, and 4.15.

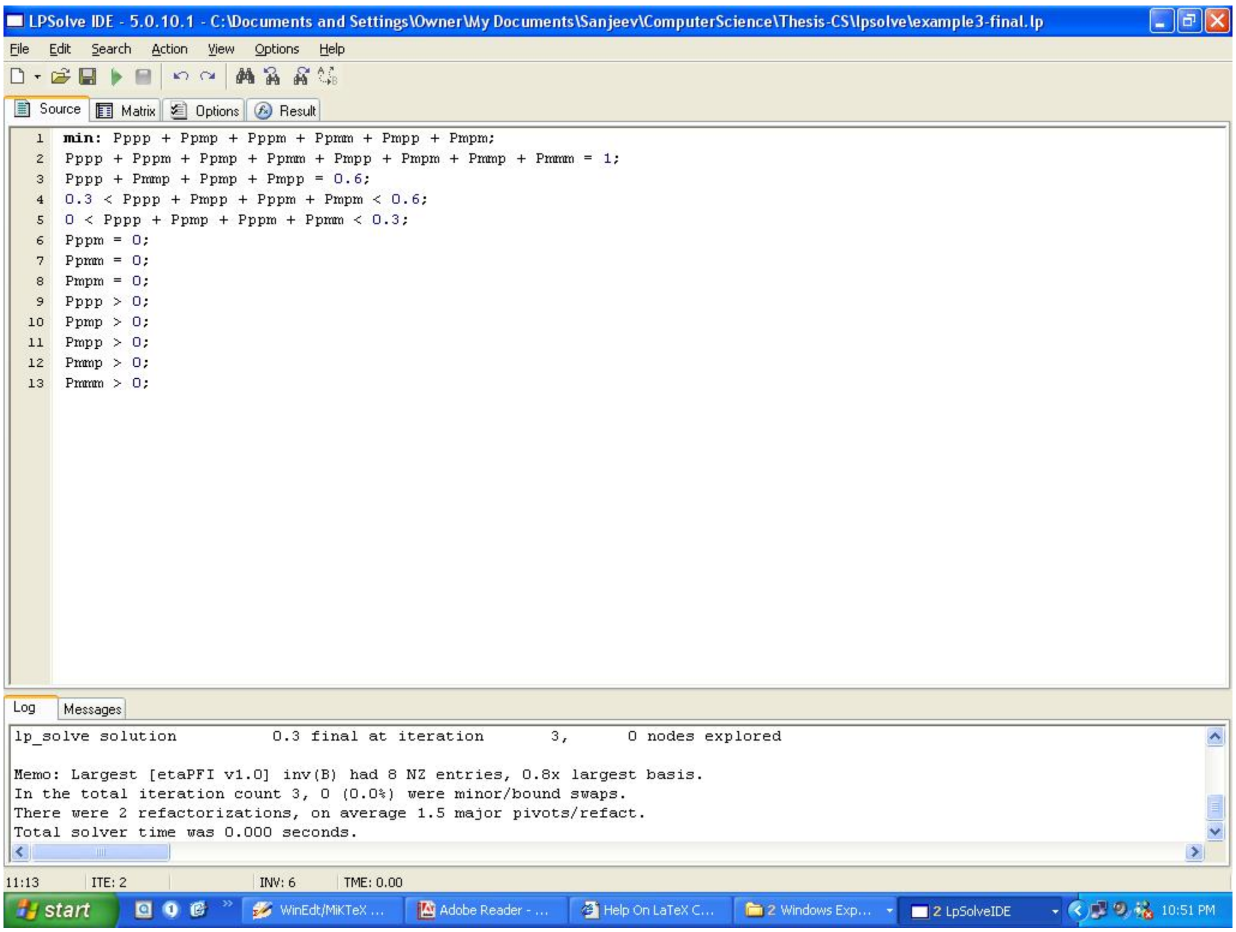

Figure 4.10: $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$: Minimization - Input/log Window

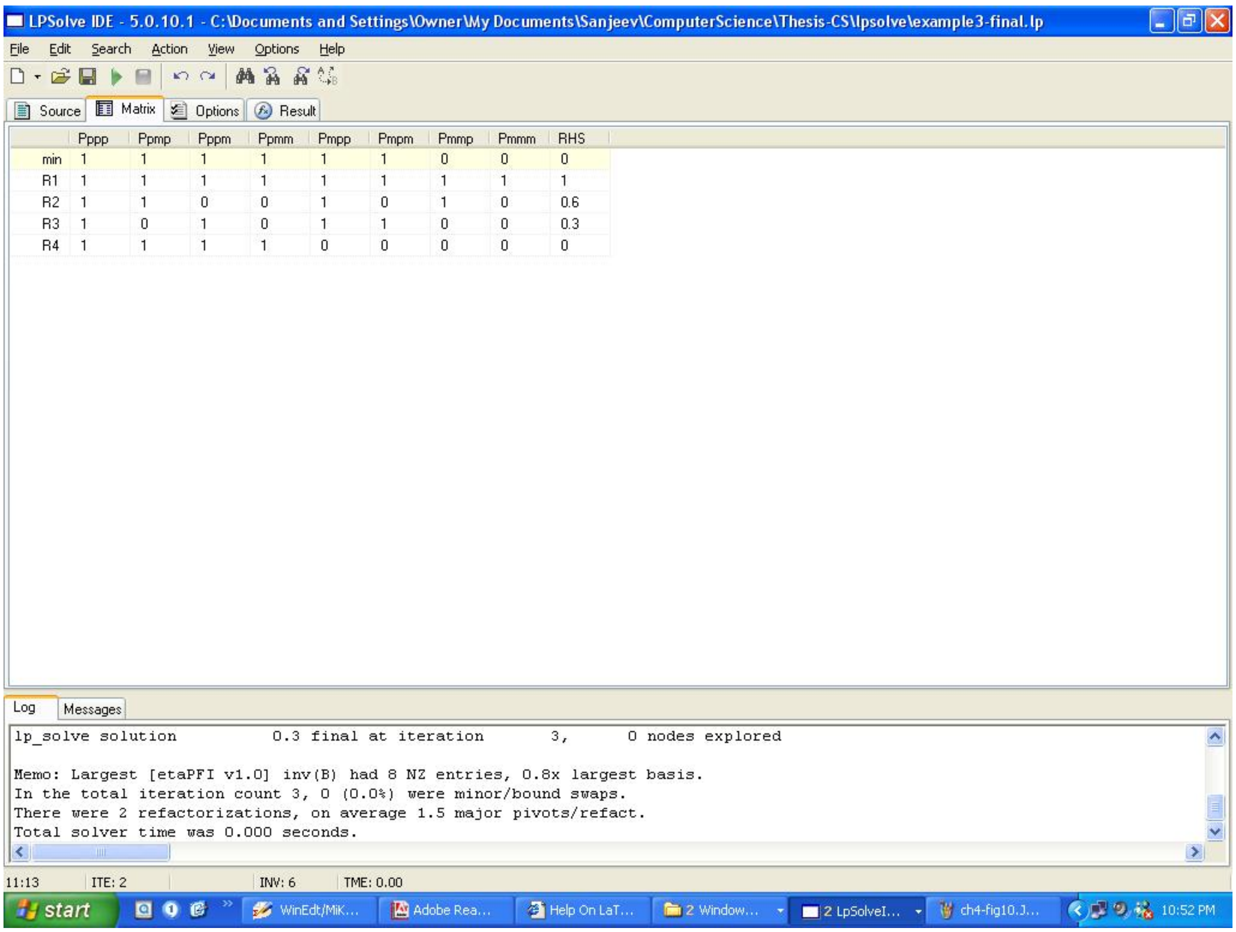

Figure 4.11: $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$: Minimization - Matrix Window

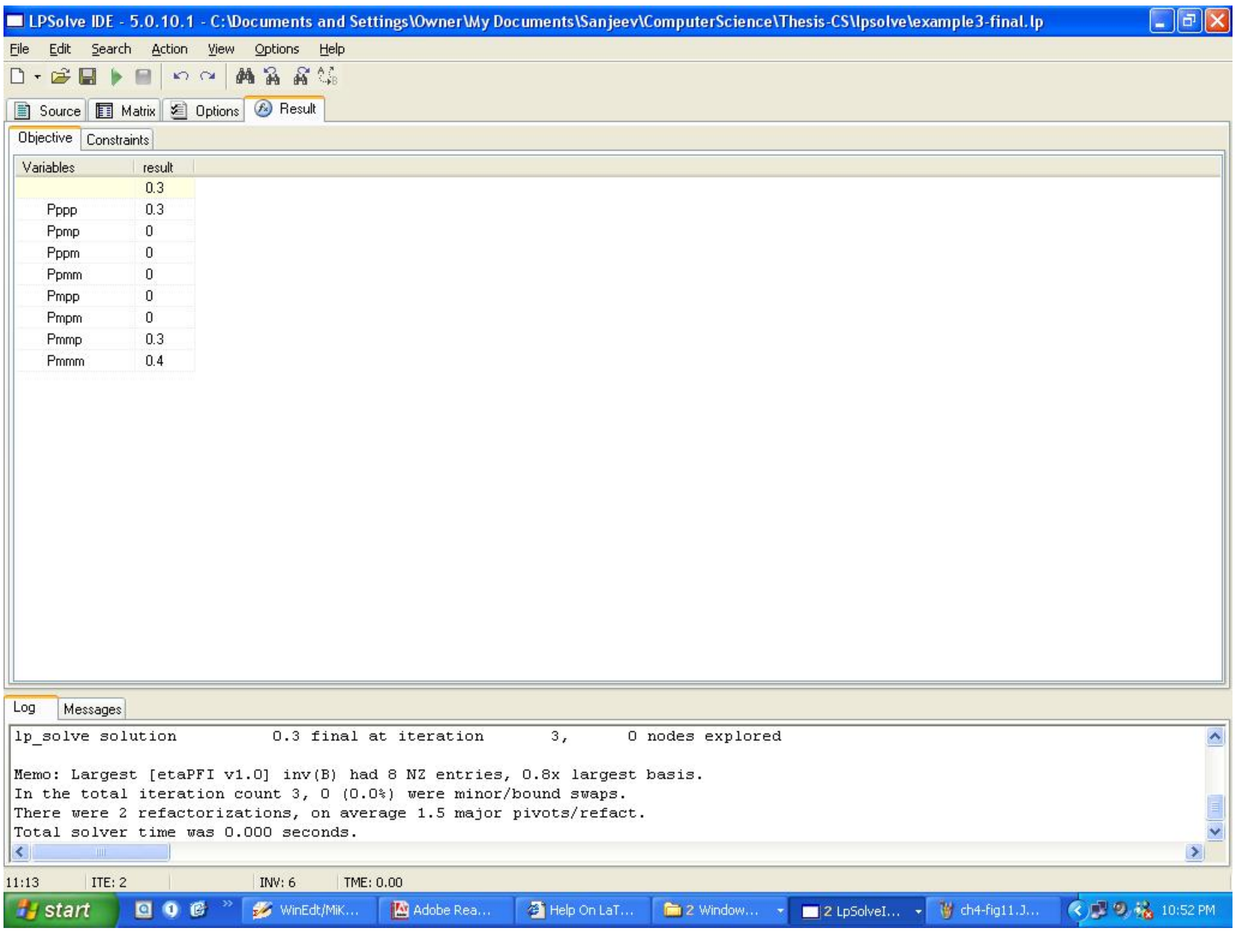

Figure 4.12: $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$: Minimization - Result Window

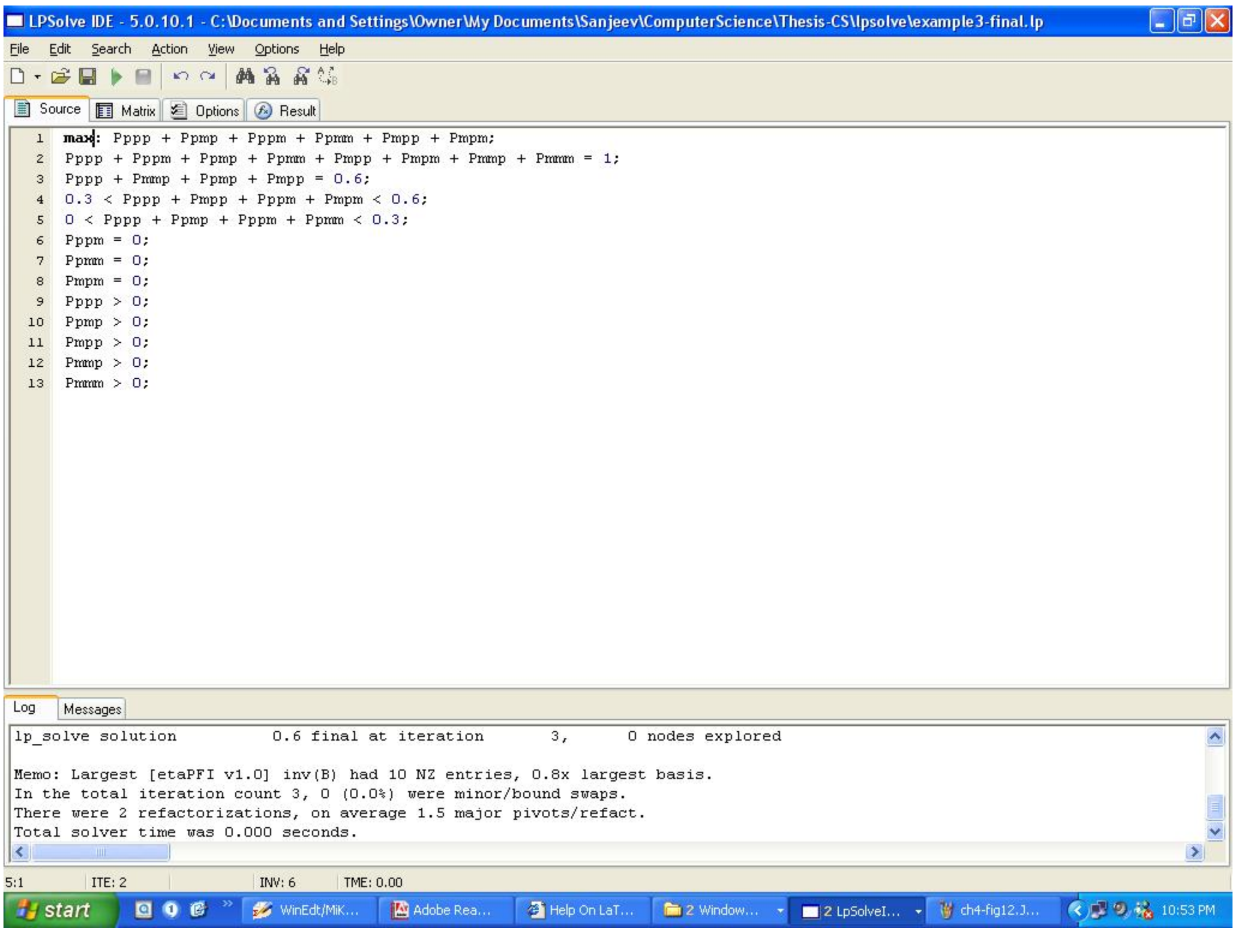

Figure 4.13: $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$: Maximization - Input/log Window

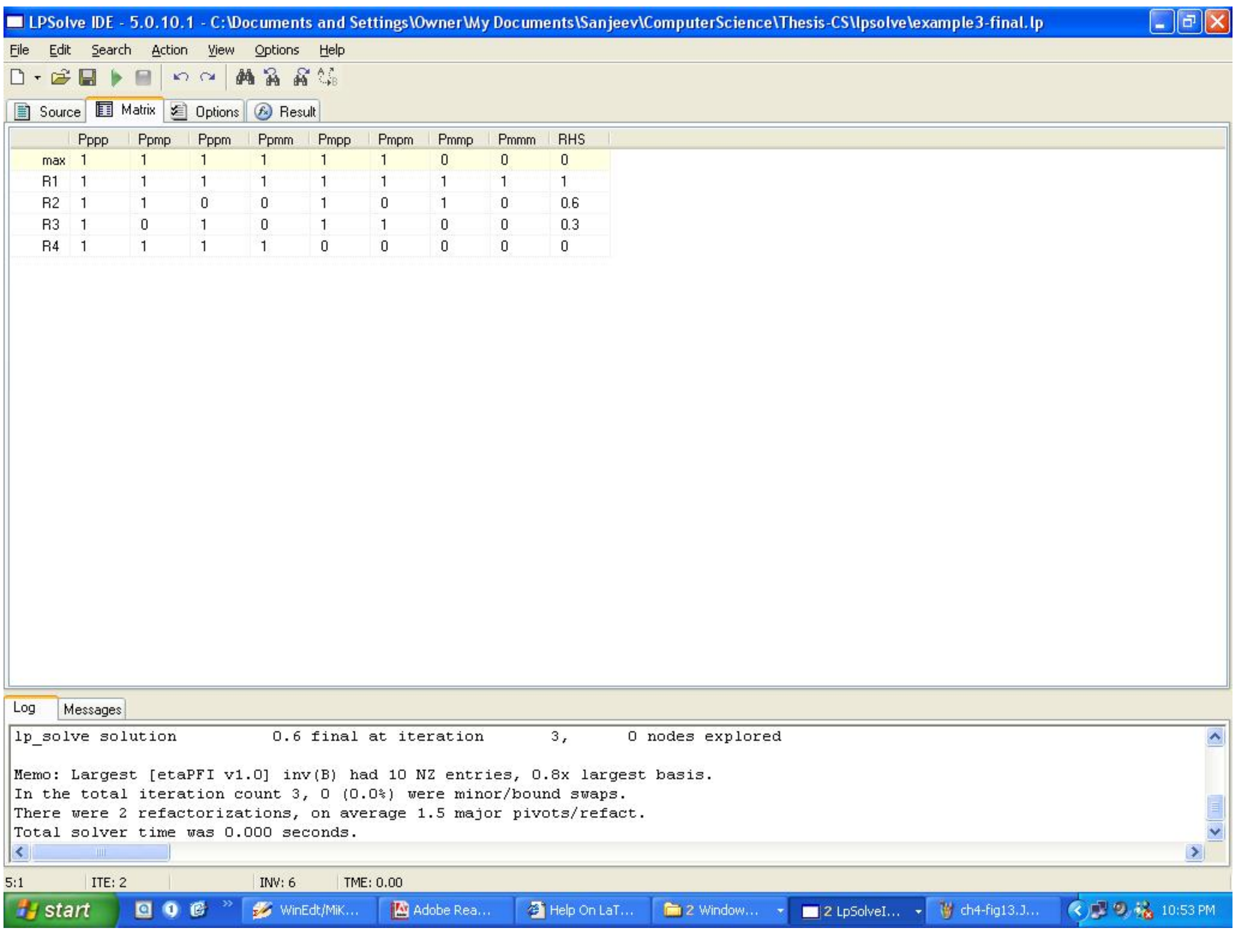

Figure 4.14: $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$: Maximization: Matrix Window

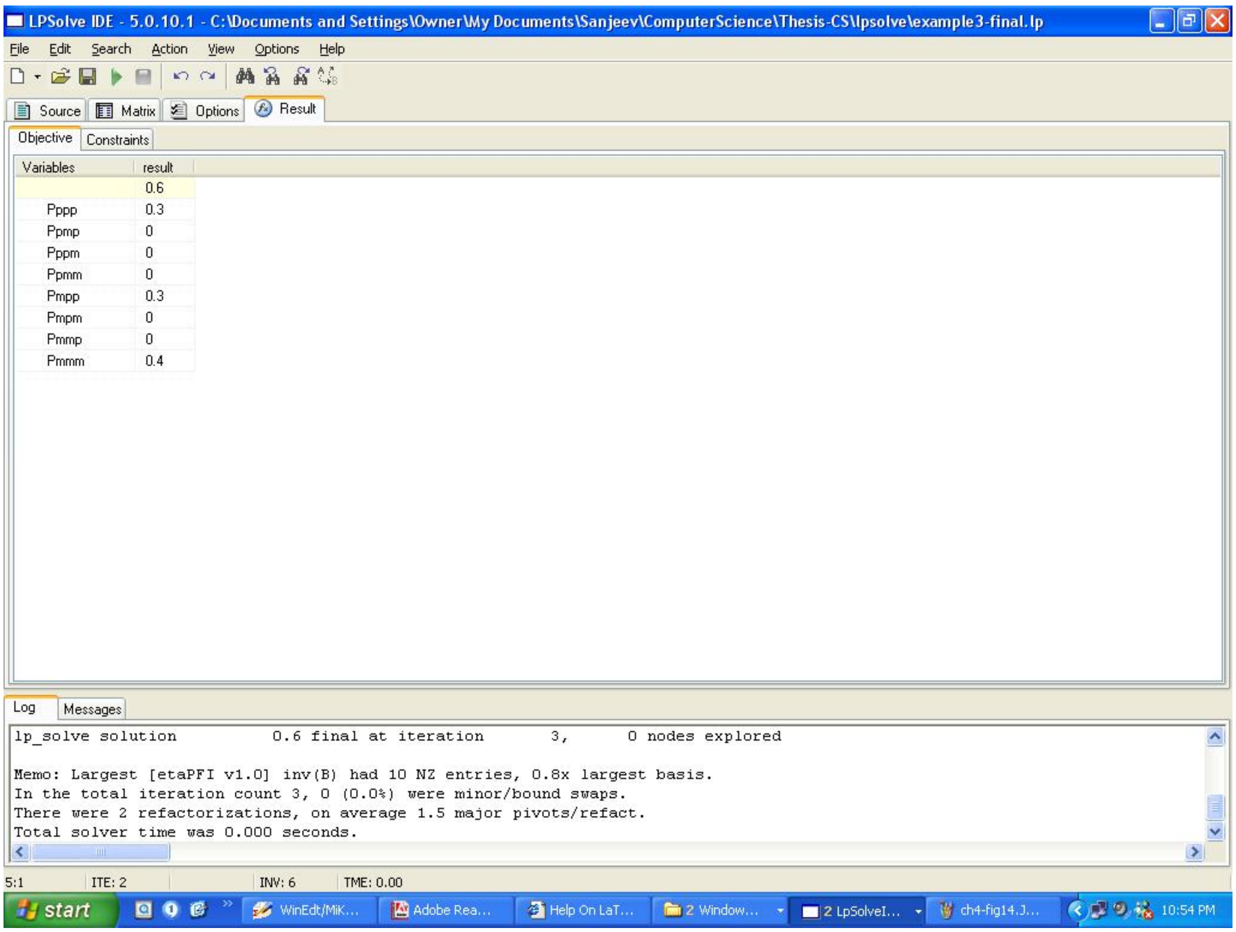

Figure 4.15: $p(A \,\&\, B) \vee p(A \,\&\, \neg B)$: Maximization: Matrix Window

Chapter 5

Conclusion

Procedures for handling uncertainty are crucial for making an effective expert system, in particular, expert systems in geoinformatics. Uncertainty of different statements is usually described as their (subjective) probability. In many real-life situations, we only have partial information about the uncertainty, so, instead of the actual probabilities, we only have *intervals* of possible values of these probabilities. Based on this information, we must find the intervals of possible values of probability $p(Q)$ of different queries Q.

In general, the problem of finding the exact bounds on the corresponding probabilities is NP-hard – simply because the propositional satisfiability problem, the known NP-hard problem, is a particular case of this problem, with probabilities equal to 1. There exist efficient heuristic approaches to handling such interval-valued uncertainty, but these heuristic approaches often lead to excess width.

In this thesis, we used the ideas of the generalized interval approach to we get narrower intervals. Specifically, we have shown that the traditional expert system approach is, in some reasonable sense, similar to the straightforward interval computation techniques. In interval computations, there exist sophisticated techniques such as affine arithmetic or, more generally, Taylor arithmetic, that lead to narrower interval estimates.

In this thesis, we show how the ideas behind affine and Taylor arithmetic can be applied to handling interval-valued probabilities in expert systems. As a result, we get a better handling of uncertainty and more effective expert systems.

The new approach is tested on several examples, and it gives satisfactory results, with narrower intervals. To perform a linear programming part of the algorithm, we use an efficient and free package `lpsolve`.

Future work could include testing and implementing this approach in different expert system applications, in particular, in geoinformatics. It is also desirable to make this approach more user-friendly, so that a user will simply describe the knowledge statements and the corresponding probabilities without the need to learn anything about the algorithm and/or about `lpsolve.` We may also want to use other ideas from interval computations to further improve our algorithms.

References

[1] C. Alsina, E. Trillas, and L. Valverde, On non-distributive logical connectives for fuzzy set theory, *Busefal*, 1980, Vol. 3, pp. 18–29.

[2] K. T. Atanassov, Intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, 1986, Vol. 20, No. 1, pp. 87–96.

[3] K. T. Atanassov, *Review and new results on intuitionistic fuzzy sets*, Preprint IM-MFAIS-1-88, Sofia, Bulgaria, 1988.

[4] K. T. Atanassov, *Four new operators on intuitionistic fuzzy sets*, Preprint IM-MFAIS-4-89, Sofia, Bulgaria, 1989.

[5] K. T. Atanassov, More on intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, 1989, Vol. 33, No. 1, pp. 37–45.

[6] K. T. Atanassov, New operations defined over the intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, 1993, Vol. 61, No. 2, pp. 131–142.

[7] K. T. Atanassov, A universal operator over intuitionistic fuzzy sets, *Compt. Rend. Acad. Bulg. Sci.*, 1993, Vol. 46, No. 1, pp. 13–15.

[8] K. T. Atanassov, Level operators on intuitionistic fuzzy sets, *BUSEFAL*, 1993, Vol. 54, pp. 4–8.

[9] K. T. Atanassov, Operations over interval valued intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, 1994, Vol. 64, pp. 159–174.

[10] K. T. Atanassov, Editor-in-Chief, Notes on Intuitionistic Fuzzy Sets, *NIFS*, International Journal, 1995–.

[11] K. T. Atanassov and G. Gargov, Interval valued intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, 1989, Vol. 31, No. 3, pp. 343–349.

[12] R. K. Bhatnagar and L. N. Kanal, Handling Uncertain Information: A Review of Numeric and Non-Numeric Methods, *Uncertainty in AI*, Elsevier North-Holland, New York, N.Y., 1986, pp. 3-26.

[13] B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh (Eds.), *Uncertainty in Knowledge Bases*, Springer-Verlag, Berlin, 1991.

[14] E. Brent, R. Benfer, L. Furbee, Reasoning under uncertainty, In: *Expert Systems*, SAGE Publications, Thousand Oaks, CA, 1991.

[15] B. G. Buchanan and E. H. Shortliffe. *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, Reading, MA, Menlo Park, CA, 1984.

[16] D. J. Buehrer, From interval probability theory to computable fuzzy first-order logic and beyond, *Proceedings of the Third IEEE Conference on Fuzzy Systems*, Orlando, FL, June 26–29, 1994, Vol. II, pp. 1428–1433.

[17] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.

[18] W. Cui and D. I. Blockley, Interval probability theory of evidential support, *International Journal of Intelligent Systems*, 1990, Vol. 5, pp. 183–192.

[19] A. P. Dawid, D. J. Spiegelhalter, S. L. Lauritzen, R. G. Cowell, Bayesian Analysis in Expert Systems, *Statistical Science*, 1993, Vol. 8, No. 3, pp. 219–283.

[20] A. P. Dempster, Upper and lower probabilities induced by a multivalued mapping, *Ann. Math. Stat.*, 1967, Vol. 38, pp. 325–339.

[21] D. Dubois, H. Prade, *Fuzzy Sets and Systems: Theory and Applications* Academic Press, N.Y., London, 1980.

[22] W. Fellner, *Probability and Profit*, Irwin, Homewood, IL, 1965.

[23] L. H. de Figueiredo and J. Stolfi, *Self-Validated Numerical Methods and Applications*, Brazilian Mathematics Colloquium monograph, IMPA, Rio de Janeiro, Brazil, 1997.

[24] G. K. Gargov and K. T. Atanassov, Two results in intuitionistic fuzzy logic, *C. R. Acad. Bulgare Sci.*, 1992, Vol. 45, No. 12, pp. 29–31.

[25] Wen-Lung Gau and D. J. Buehrer, Vague sets, *IEEE Transactions on Systems, Man, and Cybernetics*, 1993, Vol. 23, No. 2, pp. 610–614.

[26] J. Giarratano and G. Riley, *Expert Systems: Principles and Programming*, Fourth Edition, PWS Publishing Company, Boston, MA, 1998.

[27] J. Hoefkens, M. Berz, and K. Makino, Controlling the Wrapping Effect in the Solution of ODEs for Asteroids, *Reliable Computing*, 1999, Vol. 9, No. 1, pp. 21–41.

[28] P. Jaksurat, E. Freudenthal, M. Ceberio, and Vladik Kreinovich, Probabilistic Approach to Trust: Ideas, Algorithms, and Simulations, *Proceedings of the Fifth International Conference on Intelligent Technologies InTech'04*, Houston, Texas, December 2–4, 2004.

[29] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer, London, 2001.

[30] P. R. Kleindorfer, H. C. Kunreuther, and P. J. H. Schoemaker, *Decision sciences. An integrative perspective*, Cambridge University Press, 1993.

[31] V. Kreinovich, G. Shafer, J. Pearl (eds.), A review of "Uncertain Reasoning", *SIGART Bulletin*, 1992, Vol. 3, No. 4, pp. 23–27.

[32] V. Kreinovich, A. Bernat, W. Borrett, Y. Mariscal, and E. Villa, Monte-Carlo methods make Dempster-Shafer formalism feasible, In: R. R. Yager, J. Kacprzyk, and M. Pedrizzi (Eds.), *Advances in the Dempster-Shafer Theory of Evidence*, Wiley, N.Y., 1994, pp. 175–191.

[33] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational complexity and feasibility of data processing and interval computations*, Kluwer, Dordrecht, 1997.

[34] M. Ceberio, V. Kreinovich, S. Chopra, B. Ludaescher, Taylor Model-Type Techniques for Handling Uncertainty in Expert Systems, with Potential Applications to Geoinformatics, *Proceedings of the 17th World Congress of the International Association for Mathematics and Computers in Simulation IMACS'2005*, Paris, France, July 11–15, 2005.

[35] F. Lin and J. Zhao, On tight logic programs and yet another translation from normal logic programs and propositional logic, *Proceedings of the American Association for Artificial Intelligence (AAAI)*, 2003, pp. 853–864.

[36] C. H. Ling, Representation of associative functions, *Publ. Math. Debrecen*, 1965, Vol. 12, pp. 189–212.

[37] C. V. Negoita, *Expert Systems and Fuzzy Systems*, Benjamin/Cummings Publisher, Menlo Park, CA, 1985.

[38] A. Neumaier, Taylor Forms – Use and Limits, *Reliable Computing*, 2003, Vol. 9, No. 1, pp. 43–79.

[39] H. T. Nguyen, V. Kreinovich, and Q. Zuo, Interval-valued degrees of belief: applications of interval computations to expert systems and intelligent control, *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems (IJUFKS)*, 1997, Vol. 5, No. 3, pp. 317-358.

[40] H. T. Nguyen and E. A. Walker, *First Course in Fuzzy Logic*, CRC Press, Boca Raton, Florida, 1999.

[41] A. M. Norwich and I. B. Türkşen, A model for the measurement of membership and the consequences of its empirical implementation, *Fuzzy Sets and Systems*, 1984, Vol. 12, No. 1, pp. 1–25.

[42] J. R. Quinlan, Inferno: a cautious approach to uncertain inference, *Computer J.*, 1983, Vol. 26, No. 3, pp. 255–268.

[43] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Englewood Cliffs, New Jersey, 2002.

[44] B. Schweizer and A. Sklar, Associative functions and statistical triangle inequalities, *Publ. Math. Debrecen*, 1961, Vol. 8, pp. 169–186.

[45] B. Schweizer and A. Sklar, *Probabilistic metric spaces*, North Holland, N.Y., 1983.

[46] G. Shafer, *A mathematical theory of evidence*, Princeton University Press, Princeton, NJ, 1976.

[47] E. H. Shortliffe. *Computer-based medical consultation: MYCIN*, Elsevier, New York, 1976.

[48] S. J. Simoff, Interval approximate reasoning for expert systems, *Interval Computations*, 1992, No. 4, pp. 83–87.

[49] S. Smith and V. Kreinovich, In Case of Interval Uncertainty, Optimal Control is NP-Hard Even for Linear Plants, so Expert Knowledge is Needed, *Reliable Computing*, Supplement – Extended Abstracts of APIC'95: International Workshop on Applications of Interval Computations, El Paso, TX, February 23–25, 1995, pp. 190–193.

[50] M. Spies, Combination of evidence with conditional objects and its application to cognitive modeling, In: I. R. Goodman, M. M. Gupta, H. T. Nguyen, G. S. Rogers (editors), *Conditional logic in expert systems*, Elsevier, 1991, pp. 181–209.

[51] J. Stolfi and L. H. de Figueiredo, An Introduction to Affine Arithmetic, *Tendencias em Matematica Aplicada e Computacional (TEMA)*, Brazil, 2003, Vol. 4, No. 3, pp. 297–312.

[52] R. Trejo, V. Kreinovich, and C. Baral, Towards Feasible Approach to Plan Checking Under Probabilistic Uncertainty: Interval Methods, *Proc. of the 17th National Conference on Artificial Intelligence AAAI'2000*, Austin, TX, July 30–August 3, 2000, pp. 545–550.

[53] I. B. Türkşen, Measurement of membership functions and their acquisition, *Fuzzy Sets and Systems*, 1991, Vol. 50, No. 1, pp. 5–38.

[54] P. Walley and T. L. Fine, Towards a frequentist theory of upper and lower probability, *Ann. Statist.*, 1982, No. 10, pp. 741–761.

[55] P. Walley and F. M. Campello de Souza, The economic viability of solar energy systems, *Energy Systems and Policy*, 1990, No. 14, pp. 281-304.

[56] P. Walley and L. R. Pericci, Credible intervals: how credible are they?, *International Statistical Review*, 1990, No. 58.

[57] P. Walley, *Statistical reasoning with imprecise probabilities*, Chapman and Hall, N.Y., 1991.

[58] R. R. Yager, J. Kacprzyk, and M. Pedrizzi (Eds.), *Advances in the Dempster-Shafer Theory of Evidence*, Wiley, N.Y., 1994.

[59] L. A. Zadeh, Fuzzy Sets, *Inform. and Control*, 1965, Vol. 8, pp. 338–353.

[60] H. H. Zimmerman and P. Zysno, Latent connectives in human decision making, *Fuzzy Sets and Systems*, 1980, Vol.4, pp. 37–51.

Curriculum Vitae

Sanjeev Chopra was born on November 14, 1977 in New Delhi, India. He received his Bachelor's degree from the University of Madras, India.

His interest in computer science spurred him onto pursuing a degree in computer science at the University of Texas at El Paso. His focus in computer science is on the theoretical foundations of computer science, computer networks, and software engineering.

Sanjeev is currently working as the Assistant Director of the Entering Student Program at the University of Texas at El Paso.

Present address: 212 W. Schuster Ave, Apt. 203
El Paso, Texas 79902

This thesis was typed by Sanjeev Chopra.