FAST ALGORITHMS FOR COMPUTING STATISTICS UNDER INTERVAL UNCERTAINTY, WITH APPLICATIONS TO COMPUTER SCIENCE AND TO ELECTRICAL AND COMPUTER ENGINEERING

GANG XIANG

Department of Computer Science

APPROVED:

Vladik Kreinovich, Chair, Ph.D.

Martine Ceberio, Ph.D.

Luc Longpré, Ph.D.

Scott A. Starks, Ph.D.

Pablo Arenaz, Ph.D.
Dean of the Graduate School

This dissertation is dedicated to my deeply loved grandfather who passed away in 2003.

His wish to pursue a graduate study could not be fulfilled due to World War II.

And to my parents, my wife Qianyin and my son Kevin for their great love.

FAST ALGORITHMS FOR COMPUTING STATISTICS UNDER INTERVAL UNCERTAINTY, WITH APPLICATIONS TO COMPUTER SCIENCE AND TO ELECTRICAL AND COMPUTER ENGINEERING

by

GANG XIANG, B.E.

DISSERTATION

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

DOCTOR OF PHILOSOPHY

Department of Computer Science

THE UNIVERSITY OF TEXAS AT EL PASO

December 2007

Acknowledgements

I would like to express my deep-felt gratitude to my advisor, Dr. Vladik Kreinovich of the Department of Computer Science at The University of Texas at El Paso, for his advice, encouragement, enduring patience and constant support.

I also wish to thank the other members of my committee, Dr. Martine Ceberio and Dr. Luc Longpré from the Department of Computer Science and Dr. Scott A. Starks from the Department of Electrical and Computer Engineering. Their suggestions, comments and additional guidance were invaluable to the completion of this work.

Additionally, I want to thank The University of Texas at El Paso Computer Science Department professors and staff for all their hard work and dedication, providing me the means to complete my degree and prepare for a career as a computer scientist. I am especially thankful to Dr. Ann Q. Gates for her valuable nurturing advise and continuous support, to Dr. François Modave for involving me in research and sponsoring my trip to an important international conference, to Dr. David Novick for patiently helping me work on my research presentations and descriptions during his Research Methods class, to Dr. Michela Taufer for her enthusiastic help in bioinformatics-related research topics, and to many many others.

My research was partly supported by the grants from the National Science Foundation, from the Army Research Laboratory, and from Texas Department of Transportation. I was also supported by the El Paso Research and Extension Center of the Texas A&M University System, and by a travel grant from the University of Texas at El Paso Graduate School. I am very thankful to researchers who supported me in these applied research efforts, especially to Dr. James Bohm (Army Research Lab), to Dr. Yi-Chang Chiu (University of Arizona), to Dr. Reuy Long (Kelvin) Cheu (UTEP Transportation Center), to Dr. Patrick S. Debroux (Army Research Lab), to Dr. Rong Pan (Arizona State University), and to Dr. Zhuping Shen (Texas A&M University).

Last but not the least, I must thank my dear wife for putting up with me during the development of this work with continuing, loving support and no complaint. I do not have the words to express all my feelings here, only that I love you, Qianyin!

NOTE: This thesis was submitted to my Supervising Committee on July 16, 2007.

Abstract

In many engineering applications, we have to combine probabilistic and interval uncertainty. For example, in environmental analysis, we observe a pollution level $x(t)$ in a lake at different moments of time t, and we would like to estimate standard statistical characteristics such as mean, variance, autocorrelation, correlation with other measurements. In environmental measurements, we often only measure the values with interval uncertainty. We must therefore modify the existing statistical algorithms to process such interval data.

In this dissertation, we will provide a survey of known algorithms for computing various statistics under interval uncertainty and their computational complexity, a description of new algorithms, and the applications of the new algorithms, including applications to the seismic inverse problem in geosciences, to chip design in computer engineering, and to radar data processing.

Table of Contents

List of Tables

Chapter 1

Formulation of the Problem

1.1 Computing Statistics is Important

In many engineering applications, we are interested in computing statistics. For example, in environmental analysis, we observe a pollution level $x(t)$ in a lake at different moments of time t, and we would like to estimate standard statistical characteristics such as mean, variance, autocorrelation, correlation with other measurements.

For each of these characteristics C, there is an expression $C(x_1, \ldots, x_n)$ that enables us to provide an estimate for C based on the observed values $x_1, \ldots, x_n$. For example:

- a reasonable statistic for estimating the mean value of a probability distribution is the population average $E(x_1, \ldots, x_n) = \dfrac{1}{n} \cdot (x_1 + \ldots + x_n)$;
- a reasonable statistic for estimating the variance V is the population variance

$$V(x_1, \ldots, x_n) = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E)^2,$$

where $E \stackrel{\text{def}}{=} \dfrac{1}{n} \cdot \sum_{i=1}^{n} x_i$.

Comment. The population variance is often computed by using an alternative formula $V = M - E^2$, where $M = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} x_i^2$ is the population second moment.

Comment. In many practical situations, we are interested in an *unbiased* estimate of the population variance

$$V_u(x_1, \ldots, x_n) = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - E)^2.$$

In this dissertation, we will describe how to estimate V under interval uncertainty; since $V_u = \dfrac{n}{n-1} \cdot V$, we can easily transform estimates for V into estimates for V_u.

1.2 Interval Uncertainty

In environmental measurements, we often only measure the values with interval uncertainty. For example, if we did not detect any pollution, the pollution value v can be anywhere between 0 and the sensor's detection limit DL. In other words, the only information that we have about v is that v belongs to the interval $[0, DL]$; we have no information about the probability of different values from this interval.

Another example: to study the effect of a pollutant on the fish, we check on the fish daily; if a fish was alive on Day 5 but dead on Day 6, then the only information about the lifetime of this fish is that it is somewhere within the interval $[5, 6]$; we have no information about the distribution of different values in this interval.

In non-destructive testing, we look for outliers as indications of possible faults. To detect an outlier, we must know the mean and standard deviation of the normal values – and these values can often only be measured with interval uncertainty; see, e.g., [167, 175]. In other words, often, we know the result $\widetilde{x}$ of measuring the desired characteristic x, and we know the upper bound Δ on the absolute value $|\Delta x|$ of the measurement error $\Delta x \stackrel{\text{def}}{=} \widetilde{x} - x$ (this upper bound is provided by the manufacturer of the measuring instrument), but we have no information about the probability of different values $\Delta x \in [-\Delta, \Delta]$. In such situations, after the measurement, the only information that we have about the true value x of the measured quantity is that this value belongs to interval $[\widetilde{x} - \Delta, \widetilde{x} + \Delta]$.

In geophysics, outliers should be identified as possible locations of minerals; the importance of interval uncertainty for such applications was emphasized in [157, 158]. Detecting outliers is also important in bioinformatics [187].

In bioinformatics and bioengineering applications, we must solve systems of linear equations in which coefficients come from experts and are only known with interval uncertainty; see, e.g., [215].

In biomedical systems, statistical analysis of the data often leads to improvements in medical recommendations; however, to maintain privacy, we do not want to use the exact

values of the patient's parameters. Instead, for each parameter, we select fixed values, and for each patient, we only keep the corresponding range. For example, instead of keeping the exact age, we only record whether the age is between 0 and 10, 10 and 20, 20 and 30, etc. We must then perform statistical analysis based on such interval data; see, e.g., [111, 209].

1.3 Estimating Statistics Under Interval Uncertainty: a Problem

In all such cases, instead of the true values $x_1, \ldots, x_n$, we only know the intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1], \ldots, \mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$ that contain the (unknown) true values of the measured quantities. For different values $x_i \in \mathbf{x}_i$, we get, in general, different values of the corresponding statistical characteristic $C(x_1, \ldots, x_n)$. Since all values $x_i \in \mathbf{x}_i$ are possible, we conclude that all the values $C(x_1, \ldots, x_n)$ corresponding to $x_i \in \mathbf{x}_i$ are possible estimates for the corresponding statistical characteristic. Therefore, for the interval data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, a reasonable estimate for the corresponding statistical characteristic is the range

$$C(\mathbf{x}_1, \ldots, \mathbf{x}_n) \stackrel{\text{def}}{=} \{C(x_1, \ldots, x_n) \,|\, x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}.$$

We must therefore modify the existing statistical algorithms so that they compute, or bound these ranges. This is the problem that we will be solving in this dissertation.

1.4 This Problem is a Part of a General Problem

The above range estimation problem is a specific problem related to a combination of interval and probabilistic uncertainty. Such problems – and their potential applications – have been described, in a general context, in the monographs [123, 203]; for further developments, see, e.g., [18, 19, 20, 21, 60, 66, 127, 139, 180, 181, 206] and references therein.

1.5 Outline of the Dissertation

In Chapter 2, we analyze the problem of statistical analysis under interval uncertainty: we describe traditional methods for such analysis and the limitations of these methods, thus explaining why new methods are needed. Since in general, the corresponding problems are computationally difficult (NP-hard), we can only expect efficient algorithms for particular cases of these problems. In Chapter 3, we describe practically important cases of the interval-based statistical analysis for which we would like to design efficient algorithms. In Chapter 4, for different statistical characteristics, we describe the efficient algorithms that were known before our research, and new algorithms that we propose in this dissertation. Chapter 5 describes practical applications of these algorithms. Conclusions are given in Chapter 6.

Chapter 2

Analysis of the Problem

In the previous chapter, we formulated our main problem: computing statistical characteristics under interval uncertainty. We also mentioned that this problem can be viewed as a particular case of the general problem of interval computations. In this chapter, we overview the basic techniques of interval computations, use these techniques to compute statistics under interval uncertainty, and illustrate the limitation of this usage. The main conclusion of this chapter is that while sometimes these generic interval techniques work well, in general, we need to design new algorithms specifically tailored to computing statistics.

2.1 Mean

Let us start our discussion with the simplest possible characteristic: the mean. The arithmetic average E is a monotonically increasing function of each of its n variables $x_1, \ldots, x_n$, so its smallest possible value $\underline{E}$ is attained when each value x_i is the smallest possible ($x_i = \underline{x}_i$) and its largest possible value is attained when $x_i = \overline{x}_i$ for all i. In other words, the range $\mathbf{E}$ of E is equal to $[E(\underline{x}_1, \ldots, x_n), E(\overline{x}_1, \ldots, \overline{x}_n)]$. In other words, $\underline{E} = \frac{1}{n} \cdot (\underline{x}_1 + \ldots + \underline{x}_n)$ and $\overline{E} = \frac{1}{n} \cdot (\overline{x}_1 + \ldots + \overline{x}_n)$.

2.2 Variance: Computing the Exact Range is Difficult

Another widely used statistic is the variance. In contrast to the mean, the dependence of the variance V on x_i is not monotonic, so the above simple idea does not work. Rather

surprisingly, it turns out that the problem of computing the exact range for the variance over interval data is, in general, NP-hard [64, 63] which means, crudely speaking, that the worst-case computation time grows exponentially with n. Specifically, computing the upper endpoint $\overline{V}$ of the range $[\underline{V}, \overline{V}]$ is NP-hard. Moreover, if we want to compute the variance range or $\overline{V}$ with a given accuracy ε, the problem is still NP-hard. (For a more detailed description of NP-hardness in relation to interval uncertainty, see, e.g., [110].)

2.3 Linearization

From the practical viewpoint, often, we may not need the exact range, we can often use approximate linearization techniques. For example, when the uncertainty comes from measurement errors Δx_i, and these errors are small, we can ignore terms that are quadratic (and of higher order) in Δx_i and get reasonable estimates for the corresponding statistical characteristics. In general, in order to estimate the range of the statistic $C(x_1, \ldots, x_n)$ on the intervals $[\underline{x}_1, \overline{x}_1]$, $\ldots$, $[\underline{x}_n, \overline{x}_n]$, we expand the function C in Taylor series at the midpoint $\widetilde{x}_i \stackrel{\text{def}}{=} (\underline{x}_i + \overline{x}_i)/2$ and keep only linear terms in this expansion. As a result, we replace the original statistic with its linearized version $C_{\text{lin}}(x_1, \ldots, x_n) = C_0 - \sum\limits_{i=1}^{n} C_i \cdot \Delta x_i$, where $C_0 \stackrel{\text{def}}{=} C(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, $C_i \stackrel{\text{def}}{=} \dfrac{\partial C}{\partial x_i}(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, and $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$. For each i, when $x_i \in [\underline{x}_i, \overline{x}_i]$, the difference Δx_i can take all possible values from $-\Delta_i$ to Δ_i, where $\Delta_i \stackrel{\text{def}}{=} (\overline{x}_i - \underline{x}_i)/2$. Thus, in the linear approximation, we can estimate the range of the characteristic C as $[C_0 - \Delta, C_0 + \Delta]$, where $\Delta \stackrel{\text{def}}{=} \sum\limits_{i=1}^{n} |C_i| \cdot \Delta_i$.

In particular, if we take, as the statistic, the population variance $C = V$, then $C_i = \dfrac{\partial V}{\partial x_i} = \dfrac{2}{n} \cdot (\widetilde{x}_i - \widetilde{E})$, where $\widetilde{E}$ is the average of the midpoints $\widetilde{x}_i$, and $C_0 = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} (\widetilde{x}_i - \widetilde{E})^2$ is the variance of the midpoint values $\widetilde{x}_1, \ldots, \widetilde{x}_n$. So, for the variance, $\Delta = \dfrac{2}{n} \cdot \sum\limits_{i=1}^{n} |\widetilde{x}_i - \widetilde{E}| \cdot \Delta_i$.

It is worth mentioning that for the variance, the ignored quadratic term is equal to $\dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} (\Delta x_i)^2 - (\Delta E)^2$, where $\Delta E \stackrel{\text{def}}{=} \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} \Delta x_i$, and therefore, can be bounded by 0 from below and by $\Delta^{(2)} \stackrel{\text{def}}{=} \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} \Delta_i^2$ from above. Thus, the interval $[V_0 - \Delta, V_0 + \Delta + \Delta^{(2)}]$ is a

guaranteed enclosure for $\mathbf{V}$.

2.4 Linearization is Not Always Acceptable

In some cases, linearized estimates are not sufficient: the intervals may be wide so that quadratic terms can no longer be ignored, and/or we may be in a situation where we want to guarantee that, e.g., the variance does not exceed a certain required threshold. In such situations, we need to get the exact range – or at least an enclosure for the exact range.

Since, even for as simple a characteristic as variance, the problem of computing its exact range is NP-hard, we cannot have a feasible-time algorithm that always computes the exact range of these characteristics. Therefore, we must look for the reasonable classes of problems for which such algorithms are possible. Let us analyze what such classes can be.

2.5 For this Problem, Traditional Interval Methods Sometimes Lead to Excess Width

Let us show that for this problem, traditional interval methods sometimes lead to excess width.

2.5.1 Straightforward Interval Computations

Historically the first method for computing the enclosure for the range is the method which is sometimes called "straightforward" interval computations. This method is based on the fact that inside the computer, every algorithm consists of elementary operations (arithmetic operations, min, max, etc.). For each elementary operation $f(a, b)$, if we know the intervals $\mathbf{a}$ and $\mathbf{b}$ for a and b, we can compute the exact range $f(\mathbf{a}, \mathbf{b})$. The corresponding formulas form the so-called *interval arithmetic*. In straightforward interval computations, we repeat

the computations forming the program f step-by-step, replacing each operation with real numbers by the corresponding operation of interval arithmetic. It is known that, as a result, we get an enclosure for the desired range.

For the problem of computing the range of finite population average, as we have mentioned, straightforward interval computations lead to exact bounds. The reason: in the above formula for E, each interval variable only occurs once [79].

For the problem of computing the range of finite population variance, the situation is somewhat more difficult, because in the expression (1.1), each variable x_i occurs several times: explicitly, in $(x_i - E)^2$, and explicitly, in the expression for E. In this cases, often, dependence between intermediate computation results leads to excess width of the results of straightforward interval computations. Not surprisingly, we do get excess width when applying straightforward interval computations to the formula (1.1).

For example, for $\mathbf{x}_1 = \mathbf{x}_2 = [0, 1]$, the actual $V = (x_1 - x_2)^2/4$ and hence, the actual range $\mathbf{V} = [0, 0.25]$. On the other hand, $\mathbf{E} = [0, 1]$, hence

$$\frac{(\mathbf{x}_1 - \mathbf{E})^2 + (\mathbf{x}_2 - \mathbf{E})^2}{2} = [0, 1] \supset [0, 0.25].$$

It is worth mentioning that there are other formulas one can use to compute the variance of a finite population: e.g., the formula

$$V = \frac{1}{n} \sum_{i=1}^{n} x_i^2 - E^2.$$

In this formula too, each variable x_i occurs several times, as a result of which we get excess width: for $\mathbf{x}_1 = \mathbf{x}_2 = [0, 1]$, we get $\mathbf{E} = [0, 1]$ and

$$\frac{\mathbf{x}_1^2 + \mathbf{x}_2^2}{2} - \mathbf{E}^2 = [-1, 1] \supset [0, 0.25].$$

Unless there is a general formula for computing the variance of a finite population in which each interval variable only occurs once, then without using a numerical algorithm (as contrasted with am analytical expression), it is probably not possible to avoid excess interval width caused by dependence. The fact that we prove that the problem of computing

of computing the exact bound for the finite population variance is computationally difficult (in precise terms, NP-hard) makes us believe that no such formula for finite population variance is possible.

2.5.2 Centered Form

A better range is often provided by a *centered form*, in which a range $f(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of a smooth function on a box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$ is estimated as

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_n) \subseteq f(\widetilde{x}_1, \ldots, \widetilde{x}_n) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(\mathbf{x}_1, \ldots, \mathbf{x}_n) \cdot [-\Delta_i, \Delta_i],$$

where $\widetilde{x}_i = (\underline{x}_i + \overline{x}_i)/2$ is the interval's midpoint and $\Delta_i = (\underline{x}_i - \overline{x}_i)/2$ is its half-width.

When all the intervals are the same, e.g., when $\mathbf{x}_i = [0, 1]$, the centered form does not lead to the desired range. Indeed, the centered form always produced an interval centered in the point $f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$. In this case, all midpoints $\widetilde{x}_i$ are the same (e.g., equal to 0.5), hence the finite population variance $f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is equal to 0 on these midpoints. Thus, as a result of applying the centered form, we get an interval centered at 0, i.e., the interval whose lower endpoint is negative. In reality, V is always non-negative, so negative values of V are impossible.

The upper endpoint produced by the centered form is also different from the upper endpoint of the actual range: e.g., for $\mathbf{x}_1 = \mathbf{x}_2 = [0, 1]$, we have $\frac{\partial f}{\partial x_1}(x_1, x_2) = (x_1 - x_2)/2$, hence

$$\frac{\partial f}{\partial x_1}(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1 - \mathbf{x}_2}{2} = [-0.5, 0.5].$$

A similar formula holds for the derivative with respect to x_2. Since $\Delta_i = 0, 5$, the centered form leads to:

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_n) \subseteq 0 + [-0.5, 0.5] \cdot [-0.5, 0.5] + [-0.5, 0.5] \cdot [-0.5, 0.5] = [-0.5, 0.5]$$

– an excess width in comparison with the actual range $[0, 0.25]$.

2.5.3 Constraint Solving Techniques

The above methods can be often successfully supplemented by constraint solving techniques. Specifically, sometimes, in addition to the intervals of possible values of the input, we also know some constraints on the values of the desired characteristic. For example, we know that the value $V = V(x_1, \ldots, x_n)$ of the population variance should always be non-negative: $V \geq 0$, or, in set-theoretic terms, $V \in [0, \infty)$.

In the above numerical example, from the result of straightforward interval computations, we know that $V \in [-1, 1]$. Since we also know that $V \in [0, \infty)$, we can therefore conclude that V belongs to the intersection of these two sets, i.e., to the interval $[-1, 1] \cap [0, \infty) = [0, 1]$. This interval is twice narrower than the interval $[-1, 1]$ that we obtained without using constraints.

Similarly, from the result of using the centered form, we know that $V \in [-0.5, 0.5]$. Since we also know that $V \in [0, \infty)$, we can therefore conclude that V belongs to the intersection of these two sets, i.e., to the interval $[-0.5, 0.5] \cap [0, \infty) = [0, 0.5]$. This interval is twice narrower than the interval $[-0.5, 0.5]$ that we obtained without using constraints.

However, in both cases, we still have excess width in comparison with the actual range $[0, 0.25]$.

2.6 For this Problem, Traditional Optimization Methods Sometimes Require Unreasonably Long Time

A natural way to solve the problem of computing the exact range $[\underline{V}, \overline{V}]$ of the finite population variance is to solve it as a constrained optimization problem. Specifically, to find $\underline{V}$, we must find the minimum of the function (1.1) under the conditions $\underline{x}_1 \leq x_1 \leq \overline{x}_1$, $\ldots$, $\underline{x}_n \leq x_n \leq \overline{x}_n$. Similarly, to find $\overline{V}$, we must find the maximum of the function (1.1) under the same conditions.

There exist optimization techniques that lead to computing "sharp" (exact) values of

$\min(f(x))$ and $\max(f(x))$. For example, there is a method described in [93] (and effectively implemented). However, the behavior of such general constrained optimization algorithms is not easily predictable, and can, in general, be exponential in n.

For small n, this is quite doable, but for large n, the exponential computation time grows so fast that for reasonable n, it becomes unrealistically large: e.g., for $n \approx 300$, it becomes larger than the lifetime of the Universe.

2.7 We Need New Methods

Summarizing: the existing methods are either not always efficient, or do not always provide us with sharp estimates for $\underline{V}$ and $\overline{V}$. So, we need new methods.

In this dissertation, we will describe several new methods for computing the variance of the finite population, and start analyzing the problem of computing other population parameters over interval data.

Chapter 3

Reasonable Classes of Problems for Which We Can Expect Feasible Algorithms for Statistics of Interval Data

As we have mentioned at the end of Chapter 2, in general, the problem of computing statistics under interval uncertainty is NP-hard. So, we cannot expect feasible algorithms that would solve all instances of this problem. Instead, we must look for practically reasonable classes of problems for which provably feasible algorithms are possible. In this chapter, we formulate practically reasonable classes for which we have succeeded in designing feasible algorithms.

3.1 First Class: Narrow Intervals

As we have mentioned in Chapter 2, in science and engineering practice, the most widely used uncertainty estimation technique is linearization. The main idea behind linearization is that if the measurement errors Δx_i are small, we can safely ignore quadratic and higher order terms in Δx_i and replace the original difficult-to-analyze expression by its easier-to-analyze linear approximation. The accuracy of this techniques is determined by the size of the first term that we ignore, i.e., is of size $O(\Delta x_i^2)$. Thus, the narrower the intervals (i.e., the smaller the values Δx_i), the more accurate is the result of this linearization.

In real life, we want to compute the range with a certain accuracy. So, when the intervals are sufficiently accurate, the results of linearization estimation provide the desired accuracy and thus, we have a feasible algorithm for solving our problem. When the intervals become wider, we can no longer ignore the quadratic terms and thus, the problem becomes more computationally complex. In other words, when intervals are narrower, the problem of computing statistics under interval uncertainty becomes easier. It is therefore reasonable to consider the case of narrow intervals as the first case in which we can expect feasible algorithms for computing statistics of interval data.

How can we describe "narrowness" formally? The very fact that we are performing the statistical analysis means that we assume that the actual values $x_1, \ldots, x_n$ come from a probability distribution, and we want to find the statistical characteristics of this probability distribution. Usually, this distribution is continuous: normal, uniform, etc. Formally, a continuous distribution is a one for which a finite probability density $\rho(x)$ exists for every x. In this case, for every the real number a, the probability $p = \int_{a-\delta}^{a+\delta} \rho(x)\,dx$ to have a random value within an interval $[a-\delta, a+\delta]$ is approximately equal to $\rho(a) \cdot 2\delta$ and thus, tends to 0 as $\delta \to 0$. This means that for every value a, the probability to have a random value exactly equal to a is 0. In particular, this means that with probability 1, all the values $x_1, \ldots, x_n$ randomly selected from the original distribution are different.

The data intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$ contain these different values $x_1, \ldots, x_n$. When the intervals $\mathbf{x}_i$ surrounding the corresponding points x_i are narrow, these intervals do not intersect. When their widths becomes larger than the distance between the original values, the intervals start intersecting.

Thus, the ideal case of "narrow intervals" can be described as the case when no two intervals $\mathbf{x}_i$ intersect.

3.2 Second Class: Slightly Wider Intervals

Narrow intervals can be described as intervals which do not intersect at all. Namely, we have a set of (unknown) actual values $x_1 < x_2 < \ldots < x_n$, and we have intervals around each value which are so narrow that the neighboring intervals $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ do not intersect.

As the widths of the intervals increase, they start intersecting. At first, only the neighboring intervals $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ intersect, but intervals $\mathbf{x}_i$ and $\mathbf{x}_{i+2}$ still do not intersect. As the widths increase further, intervals $\mathbf{x}_i$ and $\mathbf{x}_{i+2}$ start intersecting, etc. When the intervals become very wide, all n intervals intersect.

We can therefore gauge the degree of narrowness by the number of intervals which have a common point.

Specifically, we define the case of slightly wider intervals as the situation when for some integer K, no set of K intervals has a common intersection. The case of narrow intervals correspond to $K = 2$, the next case is $K = 3$, etc. – all the way to the general case $K = n$.

As we have mentioned, the narrower the intervals, the easier the corresponding computational problem. Since the parameter K is a measure of this narrowness, it is therefore reasonable to expect that feasible algorithms exist in this case – at least for values of K which are not too large.

3.3 Third Class: Single Measuring Instrument

In Chapter 2, we have mentioned that one of the most widely used engineering techniques for dealing with measurement uncertainty is linearlization. To be able to easily compute the range $\mathbf{C}$ of a statistic C by using linearization, we must make sure not only that intervals are relatively narrow, but also that they are approximately of the same size: otherwise, if, say, Δx_i^2 is of the same order as Δx_j, we cannot meaningfully ignore Δx_i^2 and retain Δx_j. In other words, the interval data set should not combine high-accurate measurement results (with narrow intervals) and low-accurate results (with wide intervals): all measurements

should have been done by a single measuring instrument (or at least by several measuring instruments of the same type).

The traditional linearization techniques only provide us with an approximate range. However, as we will show, for some classes of problems, these approximate estimates can be refined into an efficient computation of the exact range. Because of this possibility, let us formulate, in precise terms, the class of problems for which linearization is possible, i.e., the class of problem for which all the measurements have been performed by a single measuring instrument.

How can we describe this class mathematically? A clear indication that we have two measuring instruments (MI) of different quality is that one interval is a proper subset of the other one: $[\underline{x}_i, \overline{x}_i] \subseteq (\underline{x}_j, \overline{x}_j)$.

This restriction only refers to not absolutely measurement results, i.e., to non-degenerate intervals. In addition to such interval values, we may also have machine-represented floating point values produced by very accurate measurements, so accurate that we can, for all practical purposes, consider these values exactly known. From this viewpoint, when we talk about measurements made by a single measuring instrument, we may allow degenerate intervals (i.e., exact numbers) as well.

As we will see, the absence of such pairs is a useful property that enables us to compute interval statistics faster. We will also see that this absence happens not only for measurements made by a single MI, but also in several other useful practical cases. Since this property is useful, we will give it a name.

Definition 1 *We say that a collection of intervals satisfies a* subset property *if* $[\underline{x}_i, \overline{x}_i] \not\subseteq (\underline{x}_j, \overline{x}_j)$ *for all* i *and* j *for which the intervals* $\mathbf{x}_i$ *and* $\mathbf{x}_j$ *are non-degenerate.*

3.4 Fourth Class: Several MI

After the single MI case, the natural next case is when we have several (m) MI, i.e., when our intervals are divided into several subgroups each of which has the above-described

subset property.

We have already mentioned that the case of a single MI is the easiest; the more MI we involve, the more complex the resulting problem – all the way to the general case $m = n$, when each measurement is performed by a different MI.

Since the parameter m is a measure of complexity, it is therefore reasonable to expect that feasible algorithms exist for the case of a fixed number m – at least for the values of m which are not too large.

3.5 Fifth Class: Privacy Case

In the previous text, we mainly emphasized that measurement uncertainty naturally leads to intervals. It is worth mentioning, however, that interval uncertainty may also come from other sources: e.g., from the desire to protect privacy in statistical databases. Indeed, often, we collect large amounts of data about persons – e.g., during census, or during medical experiments. Statistical analysis of this data enables us to find useful correlations between, e.g., age and effects of a certain drug, or between a geographic location and income level. Because of this usefulness, it is desirable to give researchers an ability to perform a statistical analysis of this data. However, if we simply researchers to receive answers to arbitrary queries and publish the results of their analysis, then these results may reveal the information from the databases which is private and not supposed to be disclosed.

One way to protect privacy is not to keep the exact actual values of the privacy-related quantities such as salary or age in the database. Instead, we fix a finite number of thresholds, e.g., 0, 10, 20, 30 years, and for each person, we only record the corresponding age range: from 0 to 10, or from 10 to 20, or from 20 to 30, etc. Since the actual values are not stored in the database anymore, no queries can disclose these values.

So, this idea solves the privacy problem, but it opens up another problem: how can perform statistical processing on this privacy-related interval data? Suppose that we are interested in the values of a statistical characteristic $C(x_1, \ldots, x_n)$. If we knew the actual

values $x_1, \ldots, x_n$, then we could easily compute the value of this characteristic. However, in case of privacy-related interval uncertainty, all we know is intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ of possible values of x_i. Different values $x_i \in \mathbf{x}_i$ lead, in general, to different values of $C(x_1, \ldots, x_n)$. So, a reasonable idea is to return the range of possible values of the characteristic $C(x_1, \ldots, x_n)$ when $x_i \in \mathbf{x}_i$.

From the algorithmic viewpoint, we get the same problem as with measurement-related interval uncertainty: find the range of the given characteristic $C(x_1, \ldots, x_n)$ on given intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$. The *difference* between this case and the two previous cases is that, in the first two cases, we *do not know the exact values*, while in this case, in principle, it is *possible to get the exact value*, but we do not use the exact values, because we want to protect privacy.

From the mathematical viewpoint, privacy-related intervals have the following property: they either coincide (if the value corresponding to the two patients belongs to the same range) or are different, in which case they intersect in at most point. Similarly to the above situation, we also allow exact values in addition to ranges; these values correspond, e.g., to the exact records made in the past, records that are already in the public domain.

We will call interval data with this property – that every two non-degenerate intervals either coincide or intersect in at most one point – *privacy case.*

Comment. For the privacy case, the subset property is satisfied, so algorithms that work for the subset property case work for the privacy case as well.

Comment. Sometimes, in the privacy-motivated situation, we must process interval data in which intervals come from several different "granulation" schemes. For example, to find the average salary in North America, we may combine US interval records in which the salary is from 0 to 10,000 US dollars, from 10,000 to 20,000, etc., with the Canadian interval records in which the ranges are between 0 to 10,000 Canadian dollars, 10,000 to 20,000 Canadian dollars, etc. When we transform these records to a single unit, we get two different families of intervals, each of which satisfies the subset property. Thus, to handle such situations,

we can use algorithms developed for the several MI case.

3.6 Sixth Class: Non-Detects

An important practical case is the case of non-detects. Namely, many sensors are reasonably accurate, but they have a detection limit DL – so they cannot detect any value below DL but they detect values of DL and higher with a very good accuracy.

In this case, if a sensor returns a value $\widetilde{x} \geq DL$, then this value is reasonably accurate, so we can consider it exact (i.e., a degenerate interval $[\widetilde{x}, \widetilde{x}]$). However, if the sensor does not return any signal at all, i.e., the measurement result $\widetilde{x} = 0$, then the only thing we can conclude about the actual value of the quantity is that this value is below the detection limit, i.e., that it lies in the interval $[0, DL]$.

In this case, every interval is either an exact value or a *non-detect*, i.e., an interval $[0, DL_i]$ for some real number DL_i (with possibly different detection limits for different sensors). Under this assumption, the resulting non-degenerate intervals also satisfy the subset property. Thus, algorithms that work for the subset property case work for this "non-detects" case as well.

Also, an algorithm that works for the general privacy case also works for the non-detects case when all sensors have the same detection limit DL.

In the next chapter, we will describe the known and new algorithms for computing statistics of interval data for the above classes.

Chapter 4

Computing Statistics under Interval Uncertainty: Previously Known Algorithms, Their Limitations, and New Algorithms

In this chapter, for lower and upper bounds on different statistical characteristics and for different classes of interval uncertainty problems, we describe the efficient algorithms that were known before our research, and new algorithms that we propose in this dissertation.

Each section describes a bound, a characteristic, and (if appropriate) a class of interval uncertainty problems. Within each section, we have a subsection describing what was known before, and subsection(s) describing new algorithms designed during this dissertation work.

Previously known results about the computational complexity of different statistical characteristics are summarized in the following table.

Here, E is a population mean, V is a population variance,

$$S \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E)^3$$

is population skewness, and $L \stackrel{\text{def}}{=} E - k_0 \cdot \sigma$ and $U \stackrel{\text{def}}{=} E + k_0 \cdot \sigma$ are endpoints of the confidence interval, where a parameter k_0 is usually taken as $k_0 = 2$, $k_0 = 3$, or $k_0 = 6$.

The main results of this dissertation are summarized in the following table:

Case	E	V	L, U	S
Narrow int.	$O(n)$	$O(n^2)$	$O(n^2)$	?
Slightly wider narrow int.	$O(n)$	$O(n^2)$	$O(n^2)$	?
Single MI	$O(n)$	?	?	?
Several (m) MI	$O(n)$	?	?	?
Privacy case	$O(n)$	$O(n^2)$	$O(n^2)$	?
Non-detects	$O(n)$	?	?	?
General	$O(n)$	NP-hard	NP-hard	?

Table 4.1: Computational complexity of statistical analysis of interval data: previously known results

Here, the "new case" (described later) is a generalization of the case of several MI.

4.1 Lower Bound for Variance

4.1.1 Known Results

In brief, the lower bound $\underline{V}$ can be always computed in time $O(n \cdot \log(n))$ [74].

The quadratic time algorithm for computing $\underline{V}$. Before we review the $O(n \cdot \log(n))$ time algorithm, let us first review the first feasible algorithm for computing $\underline{V}$, which costs quadratic time, i.e., it takes $O(n^2)$ computational steps (arithmetic operations or comparisons) for n interval data points $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ [64].

The algorithm is as follows:

- First, we sort all $2n$ values $\underline{x}_i$, $\overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$.

Case	E	V	L, U	S
Narrow int.	$O(n)$	$O(n)$	$O(n \cdot \log(n))$	$O(n^2)$
Slightly wider narrow int.	$O(n)$	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$	?
Single MI	$O(n)$	$O(n)$	$O(n \cdot \log(n))$	$O(n^2)$
Several (m) MI	$O(n)$	$O(n^m)$	$O(n^m)$	$O(n^{2m})$
New case	$O(n)$	$O(n^m)$	?	?
Privacy case	$O(n)$	$O(n)$	$O(n \cdot \log(n))$	$O(n^2)$
Non-detects	$O(n)$	$O(n)$	$O(n \cdot \log(n))$	$O(n^2)$
General	$O(n)$	NP-hard	NP-hard	?

Table 4.2: Computational complexity of statistical analysis of interval data: an overview including new results from this dissertation

- Second, we compute $\underline{E}$ and $\overline{E}$ and select all "small intervals" $[x_{(k)}, x_{(k+1)}]$ that intersect with $[\underline{E}, \overline{E}]$.
- For each of the selected small intervals $[x_{(k)}, x_{(k+1)}]$, we compute the ratio $E_k = S_k/N_k$, where

$$S_k \stackrel{\text{def}}{=} \sum_{i:\underline{x}_i \geq x_{(k+1)}} \underline{x}_i + \sum_{j:\overline{x}_j \leq x_{(k)}} \overline{x}_j,$$

  and N_k is the total number of such i's and j's. If $E_k \in [x_{(k)}, x_{(k+1)}]$, then we compute

$$V_k \stackrel{\text{def}}{=} \frac{1}{n} \cdot \left(\sum_{i:\underline{x}_i \geq x_{(k+1)}} (\underline{x}_i - E_k)^2 + \sum_{j:\overline{x}_j \leq x_{(k)}} (\overline{x}_j - E_k)^2 \right).$$

  If $N_k = 0$, we take $V_k \stackrel{\text{def}}{=} 0$.
- Finally, we return the smallest of the values V_k as $\underline{V}$.

Let us check whether this algorithm really takes $O(n^2)$ steps: Sorting takes

$$O(n \cdot \log(n))$$

steps. For each k, computing E_k or V_k takes $O(n)$ steps, therefore, computing all possible E_k and V_k takes $O(n^2)$ steps. Finally, finding the smallest V_k takes $O(n)$ steps. Thus, we can compute $\underline{V}$ in $O(n^2)$ steps.

Main idea behind this quadratic time algorithm. The algorithm for computing $\underline{V}$ is based on the fact that when a function V attains a minimum on an interval $[\underline{x}_i, \overline{x}_i]$, then either $\dfrac{\partial V}{\partial x_i} = 0$, or the minimum is attained at the left endpoint $x_i = \underline{x}_i$ – then $\dfrac{\partial V}{\partial x_i} > 0$, or the minimum is attained at the right endpoint $x_i = \overline{x}_i$ and $\dfrac{\partial V}{\partial x_i} < 0$. Since the partial derivative is equal to $(2/n) \cdot (x_i - E)$, we conclude that either $x_i = E$, or $x_i = \underline{x}_i > E$, or $x_i = \underline{x}_i < E$. Thus, if we know where E is located in relation to all the endpoints, we can uniquely determine the corresponding minimizing value x_i for every i: if $\overline{x}_i \le E$ then $x_i = \overline{x}_i$; if $x_i \le \underline{x}_i$, then $x_i = \underline{x}_i$; otherwise, $x_i = E$. The corresponding value E can be found from the condition that E is the average of all the selected values x_i.

So, to find the smallest value of V, we can sort all $2n$ bounds $\underline{x}_i, \overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots$; then, for each zone $[x_{(k)}, x_{(k+1)}]$, we compute the corresponding values x_i, find their variance V_k, and then compute the smallest of these variances V_k.

As we have mentioned, the corresponding value E can be found from the condition that E is the average of all the selected values x_i. If E is in the zone $[x_{(k)}, x_{(k+1)}]$, then we know all the values x_i. Let us denote the value of E as E_k if E is in the zone $[x_{(k)}, x_{(k+1)}]$, so $n \cdot E_k$ should be equal to the sum of these values:

$$n \cdot E_k = \sum_{i:\underline{x}_i \ge x_{(k+1)}} \underline{x}_i + (n - N_k) \cdot E_k + \sum_{j:\overline{x}_j \le x_{(k)}} \overline{x}_j,$$

where by N_k, we denoted the total number of such i's for which $\underline{x}_i \ge x_{(k+1)}$ and j's for which $\underline{x}_j \le x_{(k)}$.

Subtracting $(n - N_k) \cdot E$ from both sides of this equality, we conclude that $N_k \cdot E_k = S_k$, where

$$S_k \stackrel{\text{def}}{=} \sum_{i:\underline{x}_i \ge x_{(k+1)}} \underline{x}_i + \sum_{j:\overline{x}_j \le x_{(k)}} \overline{x}_j.$$

If $N_k = 0$, this means that $x_i = E_k$ for all i, so $V = 0$. If $N_k \ne 0$, then $E_k = S_k / N_k$.

Once E_k is computed, we check whether $E_k \in [x_{(k)}, x_{(k+1)}]$. If this condition is satisfied, we can now compute the corresponding variance V_k of all the selected values of x_i.

The $O(n \cdot \log(n))$ time algorithm for computing $\underline{V}$. The most time-consuming step in the above-described quadratic time algorithm is the step to compute all possible E_k and V_k, which alone takes quadratic time. There exists an improved algorithm which reduces the time in this step and only takes $O(n \cdot \log(n))$ steps overall [74].

The improved algorithm differs from the quadratic time algorithm in the way of computing all possible E_k and V_k: Once E_k is computed, we can now compute the corresponding variance V_k as $M_k - E^2$, where M_k is the second population moment:

$$M_k = \frac{1}{n} \cdot \sum_{i:\underline{x}_i \geq x_{(k+1)}} (\underline{x}_i)^2 + \frac{n - N_k}{n} \cdot E^2 + \frac{1}{n} \cdot \sum_{j:\overline{x}_j \leq x_{(k)}} (\overline{x}_j)^2,$$

i.e., $V_k = M'_k - \dfrac{N_k}{n} \cdot E^2$, where

$$M'_k \stackrel{\text{def}}{=} \frac{1}{n} \cdot \left(\sum_{i:\underline{x}_i \geq x_{(k+1)}} (\underline{x}_i)^2 + \sum_{j:\overline{x}_j \leq x_{(k)}} (\overline{x}_j)^2 \right).$$

Thus, we compute the initial values of S_k, N_k, and M'_k, i.e., S_0, N_0, and M'_0. This step takes linear time, i.e., $O(n)$ steps.

For each k, the values S_k, N_k, and M'_k differ from the previous value by only one or two terms – namely, e.g., the values i for which $\underline{x}_i \geq x_{(k)}$ but $\underline{x}_i < x_{(k+1)}$. In other words, the only change is for i for which $x_{(k)} \leq \underline{x}_i < x_{(k+1)}$. Since $x_{(k)}$ is the ordering of all lower and upper bounds, this means that $x_{(k)} = \underline{x}_i$.

Similarly, the only change in the second sum is the term for which $\overline{x}_j = x_{(k)}$.

So, each of these values $S_k, \ldots$, can be computed from the previous values $S_{k-1}, \ldots$ in a constant number of steps. Thus, the overall number of steps for computing them is linear in n. Accordingly, computing all possible E_k and V_k only takes $O(n)$ steps.

Thus, after this improvement, we can compute $\underline{V}$ in

$$O(n \cdot \log(n)) \text{ (sorting)} + O(n) \text{ (computing all possible } E_k \text{ and } V_k) +$$
$$O(n) \text{ (finding the smallest } V_k) = O(n \cdot \log(n)) \text{ steps.}$$

Comment. If two interval bounds happen to coincide, then for the corresponding k, we may have a difference of several terms between S_k and S_{k-1}. However, each of the $2n$ bounds can occur only once in this change, so the overall number of terms is still $O(n)$.

4.1.2 New Results

In brief, the lower bound $\underline{V}$ can be always computed in linear time, i.e., in $O(n)$ steps.

Linear-time algorithm for computing $\underline{V}$. The proposed algorithm is iterative. At each iteration of this algorithm, we have three sets:

- the set I^- of all the endpoints $\underline{x}_i$ and $\overline{x}_j$ for which we already know that for the optimal vector x, we have, correspondingly, $x_i \neq \underline{x}_i$ (for $\underline{x}_i$) or $x_j = \overline{x}_j$ (for $\overline{x}_j$);
- the set I^+ of all the endpoints $\underline{x}_i$ and $\overline{x}_j$ for which we already know that for the optimal vector x, we have, correspondingly, $x_i = \underline{x}_i$ (for $\underline{x}_i$) or $x_j \neq \overline{x}_j$ (for $\overline{x}_j$);
- the set I of the endpoints $\underline{x}_i$ and $\overline{x}_j$ for which we have not yet decided whether these endpoints appear in the optimal vector x.

In the beginning, $I^- = I^+ = \emptyset$ and I is the set of all $2n$ endpoints. At each iteration, we also update the values $N^- = \#(I^-)$ (where $\#(S)$ denotes the number of elements in a set S), $N^+ = \#(I^+)$, $S^- = \sum\limits_{\overline{x}_j \in I^-} \overline{x}_j$, and $S^+ = \sum\limits_{\underline{x}_i \in I^+} \underline{x}_i$. Initially, $N^- = N^+ = S^- = S^+ = 0$.

At each iteration, we do the following:

- first, we compute the median m of the set I;
- then, by analyzing the elements of the undecided set I one by one, we divide them into two subsets

$$P^- = \{x \in I : x \leq m\}; \quad P^+ = \{x \in I : x > m\};$$

we also compute $m^+ = \min\{x : x \in P^+\}$;

- we compute $e^- = S^- + \sum\limits_{\overline{x}_j \in P^-} \overline{x}_j$, $e^+ = S^+ + \sum\limits_{\underline{x}_i \in P^+} \underline{x}_i$,

$$n^- = N^- + \#\{\overline{x}_j \in P^-\}, \quad n^+ = N^+ + \#\{\underline{x}_i \in P^+\},$$

and $E = \dfrac{e^- + e^+}{n^- + n^+}$;

- if $E < m$, then we replace I^- with $I^- \cup P^-$, S^- with e^-, I with P^+, and N^- with n^-;

- if $E > m^+$, then we replace I^+ with $I^+ \cup P^+$, S^+ with e^+, I with P^-, and N^+ with n^+;

- if $m \le E \le m^+$, then we replace I^- with $I^- \cup P^-$, I^+ with $I^+ \cup P^+$, I with $\emptyset$, S^- with e^-, S^+ with e^+, N^- with n^-, and N^+ with n^+.

At each iteration, the set of undecided indices is divided in half. Iterations continue until all indices are decided, after which we return, as $\underline{V}$, the value of the population variance for the vector x for which:

- $x_j = \overline{x}_j$ for indices j for which $\overline{x}_j \in I^-$,

- $x_i = \underline{x}_i$ for indices i for which $\underline{x}_i \in I^+$, and

- $x_i = E$ for all other indices i.

Proof that the new algorithm for computing $\underline{V}$ takes linear time. At each iteration, computing median takes linear time, and all other operations with I take time t linear in the number of elements $|I|$ of I. We start with the set I of size n; on the next iteration, we have a set of size $n/2$, then $n/4$, etc. Thus, the overall computation time is $\le C \cdot (n + n/2 + n/4 + \ldots) \le C \cdot 2n$, i.e., linear in n.

Proof that the new algorithm always computes $\underline{V}$. In the description of quadratic time algorithm and $O(n \cdot \log(n))$ time algorithm, we already showed that if we sort all $2n$

endpoints into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$, then for some $k = k_{\min}$, the minimum $\underline{V}$ is attained for the vector x for which:

- for all indices j for which $\overline{x}_j \le x_{(k)}$, we have $x_j = \overline{x}_j$;
- for all indices i for which $\underline{x}_i \ge x_{(k+1)}$, we have $x_i = \underline{x}_i$;
- for all other indices, we have $x_i = E_k \stackrel{\text{def}}{=} \dfrac{S_k}{N_k}$, where

$$S_k = \sum_{j:\overline{x}_j \le x_{(k)}} \overline{x}_j + \sum_{i:\underline{x}_i \ge x_{(k+1)}} \underline{x}_i;$$

$$N_k = \#\{j : \overline{x}_j \le x_{(k)}\} + \#\{i : \underline{x}_i \ge x_{(k+1)}\}.$$

It has also been shown that for the optimal k, we have $E_k \in [x_{(k)}, x_{(k+1)}]$.

The details of the proofs can also be found in [63].

In general, the condition $x_{(k)} \le E_k = \dfrac{S_k}{N_k}$ is equivalent to

$$N_k \cdot x_{(k)} \le S_k = \sum_{j:\overline{x}_j \le x_{(k)}} \overline{x}_j + \sum_{i:\underline{x}_i \ge x_{(k+1)}} \underline{x}_i.$$

Subtracting $x_{(k)}$ from each of N_k terms in the right-hand side (RHS), and moving the sum of the resulting non-positive differences into the left-hand side (LHS), we conclude that

$$\sum_{j:\overline{x}_j \le x_{(k)}} (x_{(k)} - \overline{x}_j) \le \sum_{i:\underline{x}_i \ge x_{(k+1)}} (\underline{x}_i - x_{(k)}). \tag{4.1.1}$$

When we increase k, we get, in general, more terms in the LHS and fewer in the RHS, so LHS (non-strictly) increases, while the RHS non-strictly decreases. So, if the inequality (4.1.1) holds for some k, it holds for all smaller values of k as well. Thus, this inequality holds for all k until a certain value k_0.

Similarly, the condition $x_{(k+1)} \ge E_k = \dfrac{S_k}{N_k}$ is equivalent to

$$N_k \cdot E_{k+1} \ge \sum_{j:\overline{x}_j \le x_{(k)}} \overline{x}_j + \sum_{i:\underline{x}_i \ge x_{(k+1)}} \underline{x}_i.$$

Subtracting $x_{(k+1)}$ from each of N_k terms in RHS, and moving the sum of the resulting non-positive differences into LHS, we conclude that

$$\sum_{j:\overline{x}_j \le x_{(k)}} (x_{(k+1)} - \overline{x}_j) \ge \sum_{i:\underline{x}_i \ge x_{(k+1)}} (\underline{x}_i - x_{(k+1)}). \tag{4.1.2}$$

When we increase k, the LHS (non-strictly) increases, while the RHS non-strictly decreases. So, if the inequality (4.1.2) holds for some k, it holds for all larger values of k as well. Thus, this inequality holds for all k after a certain value l_0.

So, both conditions (4.1.1) and (4.1.2) are satisfied (which is equivalent to the condition $E_k \in [x_{(k)}, x_{(k+1)}]$) either for a single value $k_{\min}$, or for several sequential values $l_0, l_0 + 1, \ldots, k_0$. Let us show that if this condition is satisfied for several sequential values, this simply means that the same minimum $\underline{V}$ is attained for all these values. For that, it is sufficient to show that if both conditions (4.1.1) and (4.1.2) holds for k and for $k+1$, then the variance V has the same value for both k and $k+1$. Indeed, since (4.1.1) is true for $k+1$, we have

$$\sum_{j:\overline{x}_j \le x_{(k+1)}} (x_{(k+1)} - \overline{x}_j) \le \sum_{i:\underline{x}_i \ge x_{(k+2)}} (\underline{x}_i - x_{(k+1)}).$$

The LHS of this new inequality is smaller than or equal to the LHS of the inequality (4.1.2), and its RHS is larger than or equal to the RHS of the inequality (4.1.2). Thus, the only way for both inequalities to hold is when both sides are equal, i.e., when replacing $x_{(k)}$ with $x_{(k+1)}$ and replacing $x_{(k+1)}$ with $x_{(k+2)}$ does not change which endpoints are in I^- and which are in I^+ – and thus, does not change the corresponding value of the variance.

So:

- for $k < k_{\min}$, we have $E_k > x_{(k+1)}$,
- for $k > k_{\min}$, we have $E_k < x_{(k)}$, and
- for $k = k_{\min}$ (or, to be more precise, for $l_0 \le k \le k_0$), we have $x_{(k)} \le E_k \le x_{(k+1)}$.

Hence:

- if $E_k < x_{(k)}$, then we cannot have $k < k_{\min}$ and $k = k_{\min}$, hence $k > k_{\min}$;

- if $E_k > x_{(k+1)}$, then we cannot have $k > k_{\min}$ and $k = k_{\min}$, hence $k < k_{\min}$;
- if $x_{(k)} \le E_k \le x_{(k+1)}$, then we cannot have $k < k_{\min}$ and $k > k_{\min}$, hence $k = k_{\min}$.

Thus, the above algorithm finds the correct value of $k_{\min}$ and thence, the correct value of $\underline{V}$.

4.2 Upper Bound for Variance: General Case

Known result. We have already mentioned that computing $\overline{V}$ is, in general, an NP-hard problem.

How to compute the upper bound in the general case: known result. It is known that the maximum of a quadratic function on an interval is always attained at one of the endpoints. Thus, in principle, we can always compute the upper bound $\overline{V}$ in time $O(2^n)$: namely, it is sufficient to compute the variance V for all 2^n possible vectors $x = (x_1^{\varepsilon_1}, \ldots, x_n^{\varepsilon_n})$, where $\varepsilon_i \in \{-, +\}$, $x_i^- = \underline{x}_i$ and $x_i^+ = \overline{x}_i$; the largest of these 2^n values is the desired value $\overline{V}$.

A new NP-hardness result. In the original proof of NP-hardness as described in [110], we have $\widetilde{x}_1 = \ldots = \widetilde{x}_n = 0$, i.e., all midpoints are the same, only accuracies Δ_i are different. What if all the midpoints $\widetilde{x}_i$ are different? We can show that in this case, computing $\overline{V}$ is still an NP-hard problem: namely, for every n-tuple of real numbers $\widetilde{x}_1, \ldots, \widetilde{x}_n$, the problem of computing $\overline{V}$ for intervals $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ is still NP-hard.

To prove this result, it is sufficient to consider $\Delta_i = N \cdot \Delta_i^{(0)}$, where $\Delta_i^{(0)}$ are the values used in the original proof and N is a large integer (that will be selected later). In this case, we can describe $\Delta x_i = \widetilde{x}_i - x_i$ as $N \cdot \Delta x_i^{(0)}$, where $\Delta_i^{(0)} \in [-\Delta_i^{(0)}, \Delta_i^{(0)}]$. For large N, the difference between the variance corresponding to the values $x_i = \widetilde{x}_i + N \cdot \Delta x_i^{(0)}$ and N^2 times the variance of the values $\Delta x_i^{(0)}$ is bounded by a term proportional to N (and the coefficient at N can be easily bounded). Thus, the difference between $\overline{V}$ and $N^2 \cdot \overline{V}^{(0)}$ is

bounded by $C \cdot N$ for some known constant C. Hence, by computing $\overline{V}$ for sufficiently large N, we can compute $\overline{V}^{(0)}$ with a given accuracy $\varepsilon > 0$, and we already know that computing $\overline{V}^{(0)}$ with given accuracy is NP-hard. This reduction proves that our new problem is also NP-hard.

4.3 Upper Bound for Variance: Cases of Narrow Intervals and Slightly Wider Intervals

4.3.1 Known Results

Ideas behind the known results. For $\overline{V}$, we can provide an analysis of the derivatives which is similar to the analysis provided for $\underline{V}$. For $\overline{V}$, to this analysis, we can add the fact that the second derivative of V is ≥ 0, so there cannot be a maximum inside the interval $[\underline{x}_i, \overline{x}_i]$.

So, when $\overline{x}_i \leq E$, we take $x_i = \underline{x}_i$; when $E \leq \underline{x}_i$, we take $x_i = \overline{x}_i$; otherwise, we must consider both possibilities $x_i = \underline{x}_i$ and $x_i = \overline{x}_i$.

When intervals do not intersect, we thus end up with an $O(n^2)$ time algorithm for computing $\overline{V}$. It turns out that an $O(n^2)$ time algorithm is possible not only when the original intervals $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ do not intersect, but also in a more general case when the "narrowed" intervals $[x_i^-, x_i^+]$, where $x^- \stackrel{\text{def}}{=} \widetilde{x}_i - \Delta_i/n$ and $x_i^+ \stackrel{\text{def}}{=} \widetilde{x}_i + \Delta_i/n$ do not intersect, and even in the more general case when for some integer $K < n$, no sub-collection of greater than K narrowed intervals $[x_i^-, x_i^+]$ has a common intersection. [64].

The quadratic time algorithm for computing $\overline{V}$ in cases of narrow intervals and slightly wider intervals. This algorithm is as follows:

- First, we sort all $2n$ endpoints of the narrowed intervals $\widetilde{x}_i - \Delta_i/n$ and $\widetilde{x}_i + \Delta_i/n$ into a sequence $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(2n)}$. This enables us to divide the real line into

$2n+2$ segments ("small intervals") $[x_{(k)}, x_{(k+1)}]$, where we denoted $x_{(0)} \stackrel{\text{def}}{=} -\infty$ and $x_{(2n+1)} \stackrel{\text{def}}{=} +\infty$.

- Second, we compute $\underline{E}$ and $\overline{E}$ and pick all "small intervals" $[x_{(k)}, x_{(k+1)}]$ that intersect with $[\underline{E}, \overline{E}]$.
- For each of picked small interval $[x_{(k)}, x_{(k+1)}]$, for each i from 1 to n, we pick the following value of x_i:
  - if $x_{(k+1)} < \widetilde{x}_i - \Delta_i/n$, then we pick $x_i = \overline{x}_i$;
  - if $x_{(k)} > \widetilde{x}_i + \Delta_i/n$, then we pick $x_i = \underline{x}_i$;
  - for all other i, we consider both possible values $x_i = \overline{x}_i$ and $x_i = \underline{x}_i$.

  As a result, for each picked small interval, we get one or several sequences $x = (x_1, \ldots, x_n)$ (several if for some i, we consider both options $x_i = \overline{x}_i$ and $x_i = \underline{x}_i$). For each of these sequences, we check whether the average E of the selected values $x_1, \ldots, x_n$ is indeed within this small interval, and if it is, compute their variance.
- Finally, we return the largest of the computed variances as $\overline{V}$.

Proof of the above-described algorithm. Let us show the proof that the above-described algorithm computes $\overline{V}$ in quadratic time for all the cases when for some integer $K < n$, no sub-collection of greater than K narrowed intervals of $\mathbf{x}_i$ has a common intersection.

We will divide this proof (and all the following proofs) into parts; consecutive parts will be marked by 1°, 2°, etc. Some parts will be subdivided into subparts; in this case, subparts of a part 3° will be marked as 3.1°., 3.2°, etc.

1°. Let $x_1, \ldots, x_n$ be the values at which the variance attain its maximum on the box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$. If we fix the values of all the variables but one x_i, then V becomes a quadratic

function of x_i. When the function V attains maximum over $x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n$, then this quadratic function of one variable will attain its maximum on the interval $\mathbf{x}_i$ at the point x_i.

It can be shown that this quadratic function has a (global) minimum at $x_i = E_i'$, where E_i' is the average of all the values $x_1, \ldots, x_n$ except for x_i. Since this quadratic function of one variable is always non-negative, it cannot have a global maximum. Therefore, its maximum on the interval $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ is attained at one of the endpoints of this interval.

An arbitrary quadratic function of one variable is symmetric with respect to the location of its global minimum, so its maximum on any interval is attained at the point which is the farthest from the minimum. There is exactly one point which is equally close to both endpoints of the interval $\mathbf{x}_i$: its midpoint $\widetilde{x}_i$. Depending on whether the global minimum is to the left, to the right, or exactly at the midpoint, we get the following three possible cases:

1. If the global minimum E_i' is to the left of the midpoint $\widetilde{x}_i$, i.e., if $E_i' < \widetilde{x}_i$, then the upper endpoint is the farthest from E_i'. In this case, the maximum of the quadratic function is attained at its upper endpoint, i.e., $x_i = \overline{x}_i$.

2. Similarly, if the global minimum E_i' is to the right of the midpoint $\widetilde{x}_i$, i.e., if $E_i' > \widetilde{x}_i$, then the lower endpoint is the farthest from E_i'. In this case, the maximum of the quadratic function is attained at its lower endpoint, i.e., $x_i = \underline{x}_i$.

3. If $E_i' = \widetilde{x}_i$, then the maximum of V is attained at both endpoints of the interval $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$.

2°. In the third case, we have either $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$. Depending on whether x_i is equal to the lower or to the upper endpoints, we can "combine" the corresponding situations with Cases 1 and 2. As a result, we arrive at the conclusion that one of the following two situations happen:

1. either $E_i' \le \widetilde{x}_i$ and $x_i = \overline{x}_i$;

2. either $E_i' \geq \widetilde{x}_i$ and $x_i = \underline{x}_i$.

3°. Let us reformulate these conclusions in terms of the average $E_{\max}$ of the maximizing values $x_1, \ldots, x_n$.

The average E_i' can be described as

$$\frac{1}{n-1} \cdot \sum_{j \neq i} x_j,$$

via means the sum over all $j \neq i$. By definition, $\sum\limits_{j \neq i} x_j = \sum\limits_{j=1}^{n} x_j - x_i$, where $\sum\limits_{j=1}^{n} x_j$ is the sum over all possible j. By definition of $E_{\max}$, we have

$$E_{\max} = \frac{\sum\limits_{j=1}^{n} x_j}{n},$$

hence $\sum\limits_{j=1}^{n} x_j = n \cdot E_{\max}$. Therefore,

$$E_i' = \frac{n \cdot E_{\max} - x_i}{n-1}.$$

Let us apply this formula to the above three cases.

3.1°. In the first case, we have $\widetilde{x}_i \geq E_i'$. So, in terms of $E_{\max}$, we get the inequality

$$\widetilde{x}_i \geq \frac{n \cdot E_{\max} - x_i}{n-1}.$$

Multiplying both sides of this inequality by $n-1$, and using the fact that in this case, $x_i = \overline{x}_i = \widetilde{x}_i + \Delta_i$, we conclude that

$$(n-1) \cdot \widetilde{x}_i \geq n \cdot E_{\max} - \widetilde{x}_i - \Delta_i.$$

Moving all the terms but $n \cdot E_{\max}$ to the left-hand side and dividing by n, we get the following inequality:

$$E_{\max} \leq \widetilde{x}_i + \frac{\Delta_i}{n}.$$

3.2°. In the second case, we have $\widetilde{x}_i \le E_i'$. So, in terms of $E_{\max}$, we get the inequality

$$\widetilde{x}_i \le \frac{n \cdot E_{\max} - x_i}{n-1}.$$

Multiplying both sides of this inequality by $n-1$, and using the fact that in this case, $x_i = \underline{x}_i = \widetilde{x}_i - \Delta_i$, we conclude that

$$(n-1) \cdot \widetilde{x}_i \le n \cdot E_{\max} - \widetilde{x}_i + \Delta_i.$$

Moving all the terms but $n \cdot E_{\max}$ to the left-hand side and dividing by n, we get the following inequality:

$$E_{\max} \ge \widetilde{x}_i - \frac{\Delta_i}{n}.$$

4°. Parts 3.1 and 3.2 of this proof can be summarized as follows:

- In Case 1, we have $E_{\max} \le \widetilde{x}_i + \Delta_i/n$ and $x_i = \overline{x}_i$.
- In Case 2, we have $E_{\max} \ge \widetilde{x}_i - \Delta_i/n$ and $x_i = \underline{x}_i$.

Therefore:

- If $E_{\max} < \widetilde{x}_i - \Delta_i/n$, this means that we cannot be in Case 2. So we must be in Case 1 and therefore, we must have $x_i = \overline{x}_i$.
- If $E_{\max} > \widetilde{x}_i + \Delta_i/n$, this means that we cannot be in Case 1. So, we must be in Case 2 and therefore, we must have $x_i = \underline{x}_i$.

The only case when we do not know which endpoint for x_i we should choose is the case when $E_{\max}$ belongs to the narrowed interval $[\widetilde{x}_i - \Delta/n, \widetilde{x}_i + \Delta_i/n]$.

5°. Hence, once we know where $E_{\max}$ is with respect to the endpoints of all narrowed intervals, we can determine the values of all optimal x_i – except for those that are within this narrowed interval. Since we consider the case when no more than K narrowed intervals

can have a common point, we have no more than K undecided values x_i. Trying all possible combinations of lower and upper endpoints for these $\leq K$ values takes $\leq 2^K$ steps.

Thus, the overall number of steps is $O(2^K \cdot n^2)$. Since K is a constant, the overall number of steps is thus $O(n^2)$.

Comment. This computation time is quadratic in n, but it grows exponentially with K. So, when K grows, this algorithm takes more and more computation time; as we will see from the proof, it takes $O(2^K \cdot n^2)$ steps. In the worst case, when our conditions are not satisfied and $K = O(n)$ narrowed intervals have a common point, this algorithm takes $O(2^n \cdot n^2)$ computational steps.

4.3.2 New Results

The $O(n \cdot \log(n))$ time algorithm for computing $\underline{V}$ in cases of narrow intervals and slightly wider intervals. We improve the above-described quadratic time algorithm and reduce the time complexity to $O(n \cdot \log(n))$. The new algorithm also works in the above-described case when for some integer $K < n$, no sub-collection of greater than K narrowed intervals $[\widetilde{x}_i - \Delta_i/n, \widetilde{x}_i + \Delta_i/n]$ has a common intersection. Cases of narrow intervals and slightly wider intervals are subcases of this case. This $O(n \cdot \log(n))$ time algorithm is as follows:

1°. Let us first sort the lower endpoints $\widetilde{x}_i - \Delta_i/n$ of the narrowed intervals into an increasing sequence. Without losing generality, we can therefore assume that these lower endpoints are ordered in increasing order:

$$\widetilde{x}_1 - \Delta_1/n \leq \widetilde{x}_2 - \Delta_2/n \leq \ldots$$

Sorting takes time $O(n \cdot \log(n))$; see, e.g., [42].

2°. Then, similar to the previously quadratic algorithm, we sort *all* the endpoints of the narrowed intervals into a sequence $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(k)} \leq \ldots \leq x_{(2n)}$. Sorting means

that for every i, we know which element $k^-(i)$ represents the lower endpoint of the i-th narrowed interval and which element $k^+(i)$ represents the upper endpoint of the i-th narrowed interval.

This sorting also takes $O(n \cdot \log(n))$ steps.

3°. On the third stage, we produce, for each of the resulting zones $[x_{(k)}, x_{(k+1)}]$, the set C_k of all the indices i for which the i-th narrowed interval

$$[\widetilde{x}_i - \Delta_i/n, \widetilde{x}_i + \Delta_i/n]$$

contains this zone.

As we have mentioned, for each i, we know the value $k = k^-(i)$ for which $\widetilde{x}_i - \Delta_i/n = x_{(k)}$. So, for each i, we place i into the set $C_{k^-(i)}$ corresponding to the zone $[x_{(k^-(i))}, x_{(k^-(i)+1)}]$, into the set corresponding to the next zone, etc., until we reach the zone for which the upper endpoint is exactly $\widetilde{x}_i + \Delta_i/n$.

Here, we need one computational step for each new entry of i into the set corresponding to a new zone. Therefore, filling in all these sets takes as many steps as there are items in all these sets. For each of $2n+1$ zones, as we have mentioned, there are no more than K items in the corresponding set; therefore, overall, all the sets contain no more than $K \cdot (2n+1) = O(n)$ steps. Thus, this stage takes $O(n)$ time.

4°. On the fourth stage, for all integers p from 0 to n, we compute the sums

$$E_p \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{p} \underline{x}_i + \frac{1}{n} \cdot \sum_{i=p+1}^{n} \overline{x}_i;$$

$$M_p \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{p} (\underline{x}_i)^2 + \frac{1}{n} \cdot \sum_{i=p+1}^{n} (\overline{x}_i)^2.$$

We compute these values sequentially. Once we know E_p and M_p, we can compute E_{p+1} and M_{p+1} as $E_{p+1} = E_p + \underline{x}_{p+1} - \overline{x}_{p+1}$ and $M_{p+1} = M_p + (\underline{x}_{p+1})^2 - (\overline{x}_{p+1})^2$.

Transition from E_p and M_p to E_{p+1} and M_{p+1} takes a constant number of computational steps; so overall, we need $O(n)$ steps to compute all the values E_p and M_p.

5°. Finally, for each zone k, we compute the corresponding values of the variance. For that, we first find the smallest index i for which $x_{(k+1)} \le \widetilde{x}_i - \Delta_i/n$. We will denote this value i by $p(k)$.

Since the values $\widetilde{x}_i - \Delta_i/n$ are sorted, we can find this i by using bisection [42]. It is known that bisection takes $O(\log(n))$ steps, so finding such $p(k)$ for all $2n+1$ zones takes $O(n \cdot \log(n))$ steps.

Once $i \ge p(k)$, then $\widetilde{x}_i - \Delta_i/n \ge \widetilde{x}_{p(k)} - \Delta_{p(k)}/n \ge x_{(k+1)}$. So, in accordance with the above justification for the quadratic-time algorithm, we should select $x_i = \overline{x}_i$, as in the sums $E_{p(k)}$ and $M_{p(k)}$.

In accordance with the same justification, the only values $i < p(k)$ for which we may also select $x_i = \overline{x}_i$ are the values for which the i-th narrowed intervals contains this zone. These values are listed in the set C_k of no more than K such intervals. So, to find all possible values of V, we can do the following.

We then consider all subsets $s \subseteq C_k$ of the set C_k; there are no more than 2^K such subsets. For each subset s, we replace, in $E_{p(k)}$ and $M_{p(k)}$, values $\underline{x}_i$ and $(\underline{x}_i)^2$ corresponding to all $i \in s$, with, correspondingly, $\overline{x}_i$ and $(\overline{x}_i)^2$.

Each replacement means subtracting no more than K terms and then adding no more than K terms, so each computation takes no more than $2K$ steps. Once we have E and V corresponding to the subset s, we can check whether E belongs to the analyzed zone and, if yes, compute $V = M - E^2$.

For each subset, we need no more than $2K+2$ computations, so for all no more than 2^K subsets, we need no more than $(2K+2) \cdot 2^K$ computations. For a fixed K, this value does not depend on n; in other words, for each zone, we need $O(1)$ steps.

To perform this computation for all $2n+1$ zones, we need $(2n+1) \cdot O(1) = O(n)$ steps.

6°. Finally, we find the largest of the resulting values V – this will be the desired value $\overline{V}$.

Finding the largest of $O(n)$ values takes $O(n)$ steps.

Overall, we need

$$O(n \cdot \log(n)) + O(n \cdot \log(n)) + O(n) + O(n) + O(n \cdot \log(n)) + O(n) = O(n \cdot \log(n))$$

steps. Thus, we have proven that our algorithm computes $\overline{V}$ in $O(n \cdot \log(n))$ steps.

4.4 Upper Bound for Variance: New Case of Subset Property for Narrowed Intervals

Description of the new case. In the previous text, we described an $O(n \cdot \log(n))$ time algorithm for computing $\overline{V}$ for the case of narrow intervals. In the following text, we will show that there exists a linear time algorithm for computing $\overline{V}$ in case of narrow intervals.

This new algorithm also works in a more general case when no narrowed interval $[\widetilde{x}_i - \Delta_i/n, \widetilde{x}_i + \Delta_i/n]$ is a proper subinterval of the interior of another narrowed interval, i.e., when $|\widetilde{x}_i - \widetilde{x}_j| \geq \dfrac{|\Delta_i - \Delta_j|}{n}$ for all $i \neq j$.

Definition 2 *We say that a collection of intervals satisfies a* subset property for narrowed intervals *if no "narrowed interval" $[x_i^-, x_i^+]$, where $x^- \stackrel{\text{def}}{=} \widetilde{x}_i - \Delta_i/n$ and $x_i^+ \stackrel{\text{def}}{=} \widetilde{x}_i + \Delta_i/n$, is a proper subinterval of the interior of another narrowed interval, i.e., when $|\widetilde{x}_i - \widetilde{x}_j| \geq \dfrac{|\Delta_i - \Delta_j|}{n}$ for all $i \neq j$.*

This case includes the cases of narrow intervals and of a single measuring instrument as particular cases.

Comment. This class if new, so there are no known results for computing $\overline{V}$ for this class.

New result: a linear time algorithm for computing $\overline{V}$ in case of subset property for narrowed intervals. The most time-consuming step in the $O(n \cdot \log(n))$ time algorithm described in Section 4.3 is sorting, which alone takes $O(n \cdot \log(n))$ time. To further reduce the time complexity, we need discard the use of sorting.

The main idea is using a linear-time algorithm for computing the median instead of an $O(n \cdot \log(n))$ time algorithm for sorting the list.

The proposed algorithm is iterative. At each iteration of this algorithm, we have three sets:

- the set I^- of all the indices i from 1 to n for which we already know that for the optimal vector x, we have $x_i = \underline{x}_i$;
- the set I^+ of all the indices j for which we already know that for the optimal vector x, we have $x_j = \overline{x}_j$;
- the set $I = \{1, \ldots, n\} - I^- - I^+$ of the indices i for which we are still undecided.

In the beginning, $I^- = I^+ = \emptyset$ and $I = \{1, \ldots, n\}$. At each iteration, we also update the values of two auxiliary quantities $S^- \stackrel{\text{def}}{=} \sum\limits_{i \in I^-} \underline{x}_i$ and $S^+ \stackrel{\text{def}}{=} \sum\limits_{j \in I^+} \overline{x}_j$. In principle, we could compute these values by computing these sums, but to speed up computations, on each iteration, we update these two auxiliary values in a way that is faster than re-computing the corresponding two sums. Initially, since $I^- = I^+ = \emptyset$, we take $S^- = S^+ = 0$.

At each iteration, we do the following:

- first, we compute the median m of the set I (median in terms of sorting by $\widetilde{x}_i$);
- then, by analyzing the elements of the undecided set I one by one, we divide them into two subsets $P^- = \{i : \widetilde{x}_i \le \widetilde{x}_m\}$ and $P^+ = \{j : \widetilde{x}_j > \widetilde{x}_m\}$;
- we compute $e^- = S^- + \sum\limits_{i \in P^-} \underline{x}_i$ and $e^+ = S^+ + \sum\limits_{j \in P^+} \overline{x}_j$;
- if $n \cdot x_m^- < e^- + e^+$, then we replace I^- with $I^- \cup P^-$, S^- with e^-, and I with P^+;
- if $n \cdot x_m^- > e^- + e^+$, then we replace I^+ with $I^+ \cup P^+$, S^+ with e^+, and I with P^-;
- if $n \cdot x_m^- = e^- + e^+$, then we replace I^- with $I^- \cup P^-$, I^+ with $I^+ \cup P^+$, and I with $\emptyset$.

At each iteration, the set of undecided indices is divided in half. Iterations continue until all indices are decided, after which we return, as $\overline{V}$, the value of the population variance for the vector x for which $x_i = \underline{x}_i$ for $i \in I^-$ and $x_j = \overline{x}_j$ for $j \in I^+$.

Comment. If some intervals $\mathbf{x}_i$ are degenerate, i.e., $\mathbf{x}_i = [x_i, x_i]$, then we need the following modifications to the above algorithm:

- first, as the initial set I, we take the set of all indices corresponding to non-degenerate intervals;
- second, we pre-compute the sum e of all the exactly known values x_i (corresponding to degenerate intervals);
- third, on each iteration, instead of comparing the product $n \cdot x_m^-$ with the sum $e^- + e^+$, we compare the product with the sum $e^- + e^+ + e$;
- finally, when computing the population variance that will be returned in $\underline{V}$, we must include the degenerate values x_i as well.

Let us check whether the new algorithm for computing $\overline{V}$ takes linear time. At each iteration, computing median takes linear time, and all other operations with I take time t linear in the number of elements $|I|$ of I: $t \le C \cdot |I|$ for some C. We start with the set I of size n; on the next iteration, we have a set of size $n/2$, then $n/4$, etc. Thus, the overall computation time is $\le C \cdot (n + n/2 + n/4 + \ldots) \le C \cdot 2n$, i.e., linear in n.

Proof that the new linear algorithm always computes $\overline{V}$ in case of subset property for narrowed intervals. Similarly to [106], one can easily show that since no two narrowed intervals are proper subsets of one another, they can be linearly ordered in lexicographic order. In this order, we have $x_1^- \le x_2^- \le \ldots \le x_n^-$, $x_1^+ \le x_2^+ \le \ldots \le x_n^+$, and, thus, the averages $\widetilde{x}_i = (x_i^- + x_i^+)/2$ are also sorted: $\widetilde{x}_1 \le \widetilde{x}_2 \le \ldots \le \widetilde{x}_n$.

In [48], we have shown that in this sorting, the value $\overline{V}$ is attained at one of the vectors $x^{(k)} = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$, i.e., that $V = V(x^{(k)})$ for some k.

In [48], we also analyzed the change in $V(x^{(k)})$ when we replace $x^{(k)}$ with $x^{(k-1)}$, i.e., when we replace $\underline{x}_k$ with $\overline{x}_k = \underline{x}_k + 2\Delta_k$; we have shown that $V_{k-1} - V_k = \dfrac{4\Delta_k}{n} \cdot (x_k^- - E_k)$, where $E_k \stackrel{\text{def}}{=} E(x^{(k)})$.

Hence, $V_{k-1} < V_k$ if and only if $x_k^- < E_k$. Multiplying both sides of this inequality by n, we get an equivalent inequality $x_k^- < n \cdot E_k$, where $n \cdot E_k = \sum\limits_{i=1}^{k} \underline{x}_i + \sum\limits_{j=k+1}^{n} \overline{x}_j$. Similarly, $V_{k-1} > V_k$ if and only if $x_k^- > E_k$, and $V_{k-1} = V_k$ if and only if $x_k^- = E_k$.

When we go from k to $k+1$, we replace the larger value $\overline{x}_{k+1}$ in the sum $n \cdot E_k$ by a smaller value $\underline{x}_k$. Thus, the sequence $n \cdot E_k$ is strictly decreasing with k, while x_k^- is (maybe non-strictly) increasing with k.So, once we have $n \cdot x_k^- < E_k$, i.e., $V_{k-1} < V_k$, these inequalities will hold for smaller k as well. Similarly, once we have $n \cdot x_k^- > E_k$, i.e., $V_{k-1} > V_k$, these inequalities will hold for larger k as well.

Once we have $n \cdot x_k^- = E_k$, i.e., $V_{k-1} = V_k$, then we will have $V_k > V_{k+1} > \ldots$ and $V_k = V_{k-1} > V_{k-2} > \ldots$, i.e., $V_k = V_{k-1}$ will be the largest value of V.

In other words, the sequence V_k first increases ($V_{k-1} < V_k$) and then starts decreasing ($V_{k-1} > V_k$), with one or two top values.

For each m, if $V_{m-1} < V_m$ (i.e., if $n \cdot x_m^- < E_m$), this means that the value $k_{\max}$ corresponding to the maximum of V is $\le m$; hence for all the indices $\le m$, we already know that in the optimal vector x, $x_i = \underline{x}_i$. Thus, these indices can be added to the set I^-.

If $V_m > V_{m-1}$ (i.e., if $n \cdot x_m^- > E_m$), this means that the value $k_{\max}$ corresponding to the maximum of V is $> m$; hence for all the indices $> m$, we already know that in the optimal vector x, $x_i = \overline{x}_i$. Thus, these indices can be added to the set I^+.

Finally, if $V_m = V_{m-1}$ (i.e., if $n \cdot x_m^- = E_m$), then this m is where maximum is attained.

The algorithm has been justified.

4.5 Upper Bound for Variance: Case of Single Measuring Instrument (Case of Subset Property)

For the case of a single measuring instrument, no efficient algorithm for computing $\overline{V}$ were previously known.

4.5.1 First New Result

For case of subset property (= single measuring instrument), we can sort the intervals in lexicographic order: $\mathbf{x}_i \le \mathbf{x}_j$ if and only if $\underline{x}_i < \underline{x}_j$ or ($\underline{x}_i = \underline{x}_j$ and $\overline{x}_i \le \overline{x}_j$).

It can be proven that the maximum of V is always attained if for some k, the first k values x_i are equal to $\underline{x}_i$ and the next $n-k$ values x_i are equal to $\overline{x}_i$. This result is proven by contradiction: if in the maximizing vector $x = (x_1, \ldots, x_n)$, some $\overline{x}_i$ is preceding some $\underline{x}_j$, $i < j$, then we can increase V while keeping E intact – which is in contradiction with the assumption that the vector x was maximizing. Specifically, to increase V, we can do the following: if $\Delta_i \le \Delta_j$, we replace $\overline{x}_i$ with $\underline{x}_i = \overline{x}_i - 2\Delta_i$ and $\underline{x}_j$ with $\underline{x}_j + 2\Delta_i$; otherwise, we replace $\underline{x}_j$ with $\overline{x}_j = \underline{x}_j + 2\Delta_j$ and $\overline{x}_i$ with $\overline{x}_i - 2\Delta_j$.

As a result, we arrive at the following algorithm: first, we sort the intervals $[\underline{x}_i, \overline{x}_i]$ in lexicographic order; then, for $k = 0, 1, \ldots, n$, compute the value $V = M - E^2$ for the corresponding vectors $x^{(k)} = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$. When we go from a vector $x^{(k)}$ to the vector $x^{(k+1)}$, only one term changes in the vector x, so only one term changes in each of the sums E and M.

How good is this algorithm? Sorting takes $O(n \cdot \log(n))$ time; computing the initial values of E and M takes linear time $O(n)$. For each k, computing the new values of E and M takes a constant number of steps, so overall, computing all n values of E, M (and hence V) takes linear time. Thus, the overall time of this algorithm is $O(n \cdot \log(n))$.

4.5.2 Second New Result

The case of subset property is a subcase of the case of subset property for narrowed intervals. Therefore, the above-described new linear time algorithm for computing $\overline{V}$ in case of subset property for narrowed intervals can also be applied to case of subset property.

4.6 Discussion: Linear Time Algorithms vs. $O(n \cdot \log(n))$ Time Algorithms

In the previous sections, we have seen that in several cases, we have both a linear time algorithm and a $O(n \cdot \log(n))$ time algorithms for computing the same bound ($\underline{V}$ or $\overline{V}$). A natural question arises: how practical are the new linear time $O(n)$ algorithms? For which n are they better than the previously proposed $O(n \cdot \log(n))$ time algorithms for computing $\underline{V}$ or $\overline{V}$?

The answer to this question comes from the following analysis. In general, the $O(f(n))$ time means that the actual computation time is $\leq C \cdot f(n)$ for some constant $C > 0$. For the known $O(n \cdot \log(n))$ time algorithms, the constants are $C \approx 1$. As one can see from the proof, for our new algorithms, the constants are the same as for known linear time algorithm for computing the median, i.e., it is ≈ 20 [42]. Thus, the new algorithm is better when $\log_2(n) > 20$, i.e., when $n > 10^6$.

We have mentioned that in many practical applications we do need to process millions of data points; in such applications, the new linear algorithms for computing $\underline{V}$ or $\overline{V}$ are indeed faster. For smaller size databases, the $O(n \cdot \log(n))$ time algorithms are faster.

4.7 Upper Bound for Variance: Case of Several MI

For the case of several MI, no efficient algorithm for computing $\overline{V}$ was previously known. Let us describe the new algorithm.

In the case of several MI, we can similarly prove that if we sort the intervals corresponding to each MI in lexicographic order, then the maximum of V is attained when from intervals corresponding to each MI, the values x_i corresponding to this MI form a sequence $(\underline{x}_1, \ldots, \underline{x}_{k_j}, \overline{x}_{k_j+1}, \ldots, \overline{x}_{n_j})$, where n_j is the total number of intervals corresponding to the j-th MI.

Thus, to find the maximum of V, we must find the values $k_1, \ldots, k_m$ corresponding to

m MIs. For these values, $V = M - E^2$, where $M = \sum M_j$ and $E = \sum E_j$, where we denoted by E_j and M_j, the averages of, correspondingly, x_i and x_i^2, taken by using only results of j-th MI.

For each MI j, we can compute all $n_j + 1$ possible values E_j and M_j in linear time.

There are $\leq n^m$ combinations of k_is; for each combination, we need m additions to compute $E = \sum E_j$, m additions to compute $M = \sum M_j$, and a constant number of operations to compute $V = M - E^2$. Thus, overall, we need time $O(n^m)$.

4.8 Upper Bound for Variance: Cases of Privacy and Non-detects

These two cases are subcases of case of subset property, and in turn subcases of case of subset property for narrowed intervals. All the known and new algorithms described in Section 4.4 and 4.5 can be applied to these two cases.

4.9 Upper Bound for Variance: A New Algorithm Applicable to All Above-Described Cases

4.9.1 Analysis of all above-described cases

One can easily check that all the cases for which a feasible algorithm is known for computing upper bound for variance are particular cases of one of the following three cases:

- case of subset property for narrowed intervals – when no two narrowed intervals $[\widetilde{x}_i - \Delta_i/n, \widetilde{x}_i + \Delta_i/n]$ are proper subsets of each other; for this case, a linear time algorithm is possible;
- case of slightly wider narrowed intervals – when every collections of $> K$ narrowed intervals $\mathbf{X}_i$ has an empty intersection: $\mathbf{X}_{i_1} \cap \ldots \cap \mathbf{X}_{i_{c+1}} = \emptyset$; for this case, an

$O(n \cdot \log(n))$ time algorithm is possible; and

- case of $m > 1$ measuring instruments; for this case, an $O(n^m)$ time algorithm is possible.

Thus, to provide the most general case, we must describe a case that includes these three situations as subcases.

4.9.2 New result

Formulation of the new result. We will consider the case described by two parameters $m \geq 1$ and $K \geq 1$. In this case, we can divide the intervals $\mathbf{x}_i$ into m subclasses such that:

- the first $m-1$ subclasses have the property that within each subclass, no two narrowed intervals are proper subsets of each other;
- the last class either has the same property, or it has the property that every collections of $> K$ narrowed intervals from this class has an empty intersection.

Our algorithm will take time $O(n \cdot \log(n))$ when $m = 1$ and time $O(n^m)$ when $m > 1$.

One can easily check that all three above cases are indeed particular cases of the above situation. Namely, the case of subset property for narrowed intervals and the case of slightly wider narrowed intervals correspond to $m = 1$ – and moreover, $m = 1$ consists of exactly these two cases.

Main idea behind the new algorithm. One can show that the optimization selection ideas behind these three algorithms, in effect, do not change if, instead of considering all n intervals, we only consider a subset of the intervals.

For example, the arguments similar to the ones presented in [48] show that if a *subsequence* of the original sequence of intervals has the property that no two narrowed subintervals from this subsequence are proper intervals of one another, then for this subsequence, the maximum value of V is attained at one of the sequences of the type $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$.

The only (minor) difference is with the third case, since for this case, we are no longer talking about a zone that contains the mean E – just one of the zones.

Thus, we arrive at the following algorithm.

Algorithm. For $m = 1$, depending on the situation, we can use either the algorithm for the recent case or the algorithm for the third case. So, to describe the algorithm, it is sufficient to consider the case when $m \geq 2$.

By definition, the set of intervals can be divided into m groups. Within each group, we perform the appropriate sorting. Then, we know that the value $\overline{V}$ is attained when for each of the $m - 1$ subgroups, we have a sequence of the type $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$ for an appropriate $k = k_j$, and for the last group, a sequence for which:

- if $\overline{x}_i \leq x_{(k)}$, then $x_i = \underline{x}_i$;
- if $x_{(k+1)} \leq \underline{x}_i$, then $x_i = \overline{x}_i$;

for some parameter $k = k_m$.

Similar to the third case, for each combination $(k_1, \ldots, k_{m-1})$, checking all possible values of k_m takes time $O(n)$. Thus, for all $\leq n^{m-1}$ possible combinations $(k_1, \ldots, k_{m-1})$, we need to spend $O(n)$ time – to the total of $O(n^m)$.

Additional statement: often, we do not need to know which interval belongs to which subgroup. In our description of the new algorithm, we assumed that the original set of n intervals can be divided into m subsets, and that we know which interval belongs to which subset. It turns out that in the case when all m subsets have a no-proper-subset (nps) property, there is no need to explicitly describe the corresponding m subsets – it is sufficient to know that it is, in principle, possible to subdivide the original set of n intervals into m subsets with this property.

This possibility can be, in turn, described as follows. Based on the original intervals $\mathbf{x}_i$, we can form the following directed graph:

- its vertices are the original intervals, and
- an edge $\mathbf{x}_i \to \mathbf{x}_j$ is going from the interval $[\underline{x}_i, \overline{x}_i]$ to the interval $[\underline{x}_j, \overline{x}_j]$ if and only the i-th interval is a proper subset of the (interior of the) j-th one, i.e., if and only if $[\underline{x}_i, \overline{x}_i] \subseteq (\underline{x}_j, \overline{x}_j)$.

It is easy to see that this graph is acyclic – so each chain has at most n elements in it. By the *height* h of this graph, we mean the largest length of a chain $\mathbf{x}_{i_1} \to \mathbf{x}_{i_2} \to \ldots \to \mathbf{x}_{i_h}$ from this graph.

The following statement describes the relation between the height of the graph and the number of subgroups:

- if intervals can be divided into m subgroups with the no-proper-subset (nps) property, then the height of the corresponding graph is $\leq m$;
- vice versa, if the height of the corresponding graph is m, then we can (efficiently) divide the original intervals into m subgroups with the no-proper-subset property.

Indeed, if we can divide intervals into m nps subgroups, then we cannot have a chain of length $> m$: otherwise, at least two intervals from this chain will be in the same subgroups – and since every two elements from a chain are proper subsets of each other, this would violate the nps property.

Vice versa, if we have a graph of height m, then we can do the following:

- We take all elements which are not dominated by anyone else as the first subgroup. It is easy to see that this group has a nps property.
- After deleting elements from the first group, we can again consider those who are not dominated by anyone in the remaining graph – these will form the second subgroup.
- etc.

One can check that each interval $\mathbf{x}_i$ will be assigned to the group whose number k is the largest length of the chain leading to $\mathbf{x}_i$. Since the height of the graph is m, we will thus subdivide all n original intervals into to m subgroups with the nps property.

The statement has been proven.

4.10 Other Statistical Characteristics – Outlier Detection

4.10.1 Formulation of the Problem

Outlier detection is important. In many application areas, it is important to detect *outliers*, i.e., unusual, abnormal values; see, e.g., [115]. In medicine, unusual values may indicate disease; in geophysics, abnormal values may indicate a mineral deposit or an erroneous measurement result; in structural integrity testing, abnormal values may indicate faults in a structure, etc.

The traditional engineering approach to outlier detection (see, e.g., [175]) is as follows:

- first, we collect measurement results $x_1, \ldots, x_n$ corresponding to normal situations;
- then, we compute the sample average $E \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$ of these normal values and the (sample) standard deviation $\sigma = \sqrt{V}$, where $V \stackrel{\text{def}}{=} M - E^2$ and $M \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2$;
- finally, a new measurement result x is classified as an outlier if it is outside the interval $[L, U]$ (i.e., if either $x < L$ or $x > U$), where $L \stackrel{\text{def}}{=} E - k_0 \cdot \sigma$, $U \stackrel{\text{def}}{=} E + k_0 \cdot \sigma$, and $k_0 > 1$ is some pre-selected value (most frequently, $k_0 = 2$, 3, or 6).

Outlier detection under interval uncertainty. In some practical situations, we only have intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ of possible values of x_i. This happens, for example, if instead of observing the actual value x_i of the random variable, we observe the value $\widetilde{x}_i$ measured by an instrument with a known upper bound Δ_i on the measurement error; then, the actual

(unknown) value is within the interval $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. For different values $x_i \in \mathbf{x}_i$, we get different bounds L and U. Possible values of L form an interval – we will denote it by $\mathbf{L} \stackrel{\text{def}}{=} [\underline{L}, \overline{L}]$; possible values of U form an interval $\mathbf{U} \stackrel{\text{def}}{=} [\underline{U}, \overline{U}]$. In other words, we arrive at the following computation problem:

GIVEN:

- an integer $n \geq 1$;
- n intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$;
- a real number $k_0 > 1$.

COMPUTE the intervals

$$\mathbf{L} \stackrel{\text{def}}{=} \{L(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\};$$

$$\mathbf{U} \stackrel{\text{def}}{=} \{U(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\};$$

where:

$$L \stackrel{\text{def}}{=} E - k_0 \cdot \sigma, \quad U \stackrel{\text{def}}{=} E + k_0 \cdot \sigma,$$

$$E \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{n} x_i, \quad \sigma \stackrel{\text{def}}{=} \sqrt{M - E^2}, \text{ and}$$

$$M \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2.$$

How do we now detect outliers? There are two possible approaches to this question: we can detect *possible* outliers and we can detect *guaranteed* outliers:

- a value x is a possible outlier if it is located outside one of the possible k_0-sigma intervals $[L, U]$ (but is may be inside some other possible interval $[L, U]$);
- a value x is a guaranteed outlier if it is located outside all possible k_0-sigma intervals $[L, U]$.

Which approach is more reasonable depends on a possible situation:

- if our main objective is not to miss an outlier, e.g., in structural integrity tests, when we do not want to risk launching a spaceship with a faulty part, it is reasonable to look for possible outliers;
- if we want to make sure that the value x is an outlier, e.g., if we are planning a surgery and we want to make sure that there is a micro-calcification before we start cutting the patient, then we would rather look for guaranteed outliers.

The two approaches can be described in terms of the endpoints of the intervals **L** and **U**:

- A value x is guaranteed to be normal – i.e., it is not a possible outlier – if x belongs to the *intersection* of all possible intervals $[L, U]$, i.e., to the interval $[\overline{L}, \underline{U}]$.
- A value x is possibly normal – i.e., it is not a guaranteed outlier – if x belongs to the *union* of all possible intervals $[L, U]$, i.e., to the interval $[\underline{L}, \overline{U}]$.

So, to detect outliers under interval uncertainty, we must compute the bounds $\underline{L}$, $\overline{U}$, $\overline{L}$, and $\underline{U}$.

4.10.2 Known Results

Feasible algorithms for computing $\underline{U}$ and $\overline{L}$. The algorithms for computing $\underline{U}$ and $\overline{L}$, as shown in [114] are as follows:

- In both algorithms, first, we sort all $2n$ values $\underline{x}_i$, $\overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$; take $x_{(0)} = -\infty$ and $x_{(2n+1)} = +\infty$. Thus, the real line is divided into $2n + 1$ zones $(x_{(0)}, x_{(1)}]$, $[x_{(1)}, x_{(2)}]$, $\ldots$, $[x_{(2n-1)}, x_{(2n)}]$, $[x_{(2n)}, x_{(2n+1)})$.
- For each of these zones $[x_{(k)}, x_{(k+1)}]$, $k = 0, 1, \ldots, 2n$, we compute the values

$$e_k \stackrel{\text{def}}{=} \sum_{i:\underline{x}_i \ge x_{(k+1)}} \underline{x}_i + \sum_{j:\overline{x}_j \le x_{(k)}} \overline{x}_j,$$

$$m_k \stackrel{\text{def}}{=} \sum_{i:\underline{x}_i \geq x_{(k+1)}} (\underline{x}_i)^2 + \sum_{j:\overline{x}_j \leq x_{(k)}} (\overline{x}_j)^2,$$

and $n_k =$ the total number of such i's and j's. Then, we solve the quadratic equation

$$A_k - B_k \cdot \mu + C_k \cdot \mu^2 = 0,$$

where

$$A_k \stackrel{\text{def}}{=} e_k^2 \cdot (1+\alpha^2) - \alpha^2 \cdot m_k \cdot n; \quad \alpha \stackrel{\text{def}}{=} 1/k_0,$$

$$B_k \stackrel{\text{def}}{=} 2 \cdot e_k \cdot \left((1+\alpha^2) \cdot n_k - \alpha^2 \cdot n\right); \quad C_k \stackrel{\text{def}}{=} n_k \cdot \left((1+\alpha^2) \cdot n_k - \alpha^2 \cdot n\right).$$

For computing $\underline{U}$, we select only those solutions for which $\mu \cdot n_k \leq e_k$ and $\mu \in [x_{(k)}, x_{(k+1)}]$; for computing $\overline{L}$, we select only those solutions for which $\mu \cdot n_k \geq e_k$ and $\mu \in [x_{(k)}, x_{(k+1)}]$. For each selected solution, we compute the values of

$$E_k = \frac{e_k}{n} + \frac{n-n_k}{n} \cdot \mu, \quad M_k = \frac{m_k}{n} + \frac{n-n_k}{n} \cdot \mu^2,$$

$$U_k = E_k + k_0 \cdot \sqrt{M_k - (E_k)^2} \text{ or } L_k = E_k - k_0 \cdot \sqrt{M_k - (E_k)^2}.$$

- Finally, if we are computing $\underline{U}$, we return the smallest of the values U_k; if we are computing $\overline{L}$, we return the smallest of the values L_k.

Justification of the above-described algorithms. We will only prove the result for $\underline{U}$; for $\overline{L}$, the proof is practically identical.

Our proof is based on the fact that the minimum of a differentiable function of x_i on an interval $[\underline{x}_i, \overline{x}_i]$ is attained either inside this interval or at one of the endpoints. If the minimum is attained inside, the derivative $\dfrac{\partial U}{\partial x_i}$ is equal to 0; if it is attained at $x_i = \underline{x}_i$, then $\dfrac{\partial U}{\partial x_i} \geq 0$; finally, if it is attained at $x_i = \overline{x}_i$, then $\dfrac{\partial U}{\partial x_i} \leq 0$. For our function,

$$\frac{\partial U}{\partial x_i} = \frac{1}{n} + k_0 \cdot \frac{x_i - E}{\sigma \cdot n};$$

thus, $\dfrac{\partial U}{\partial x_i} = 0$ if and only if $x_i = \mu \stackrel{\text{def}}{=} E - \alpha \cdot \sigma$; similarly, the non-positiveness and non-negativeness of the derivative can be described by comparing x_i with μ. Thus:

- either $x_i \in (\underline{x}_i, \overline{x}_i)$ and $x_i = \mu$,
- or $x_i = \underline{x}_i$ and $x_i = \underline{x}_i \geq \mu$,
- or $x_i = \overline{x}_i$ and $x_i = \overline{x}_i \leq \mu$.

Hence, if we know how the value μ is located with respect to all the intervals $[\underline{x}_i, \overline{x}_i]$, we can find the optimal values of x_i:

- if $\overline{x}_i \leq \mu$, then minimum cannot be attained inside or at the lower endpoint, so it is attained when $x_i = \overline{x}_i$;
- if $\mu \leq \underline{x}_i$, then, similarly, the minimum is attained when $x_i = \underline{x}_i$;
- if $\underline{x}_i < \mu < \overline{x}_i$, then the minimum is attained when $x_i = \mu$.

Hence, to find the minimum, we will analyze how the endpoints $\underline{x}_i$ and $\overline{x}_i$ divide the real line, and consider all the resulting zones.

Let the corresponding zone $[x_{(k)}, x_{(k+1)}]$ be fixed. For the i's for which $\mu \notin (\underline{x}_i, \overline{x}_i)$, the values x_i that correspond to the minimal sample variance are uniquely determined by the above formulas.

For the i's for which $\mu \in (\underline{x}_i, \overline{x}_i)$, the selected value x_i should be equal to the same value μ. To determine this μ, we will use the fact that, by definition, $\mu = E - \alpha \cdot \sigma$, where E and σ are computed by using the same value of μ. This equation is equivalent to $E - \mu \geq 0$ and $\alpha^2 \cdot \sigma^2 = (\mu - E)^2$. Substituting the above values of x_i into the formula for the mean E and for the standard deviation σ, we get the quadratic equation for μ which is described in the algorithm. So, for each zone, we can uniquely determine the values x_i that may correspond to a minimum of U.

For the actual minimum, the value μ is inside one of these zone, so the smallest of the values U_k is indeed the desired minimum.

In this algorithm, sorting takes $O(n \cdot \log(n))$ steps (see, e.g., [42]), and the rest of the algorithm takes linear time ($O(n)$) for each of $2n + 1$ zones, i.e., the total quadratic time.

In general, computing $\underline{L}$ and $\overline{U}$ is NP-hard. To be able to detect guaranteed outliers, we must be able to compute the values $\underline{L}$ and $\overline{U}$. In general, this is an NP-hard problem [114].

Feasible algorithms for computing $\underline{L}$ and $\overline{U}$ for many reasonable situations. There are algorithms for computing $\overline{U}$ and $\underline{L}$ for many reasonable situations. Namely, there are efficient algorithms that compute $\overline{U}$ and $\underline{L}$ for the case when all the interval midpoints ("measured values") $\widetilde{x}_i \stackrel{\text{def}}{=} (\underline{x}_i + \overline{x}_i)/2$ are definitely different from each other, in the sense that the "narrowed" intervals

$$\left[\widetilde{x}_i - \frac{1+\alpha^2}{n} \cdot \Delta_i, \widetilde{x}_i + \frac{1+\alpha^2}{n} \cdot \Delta_i\right]$$

– where $\alpha = 1/k_0$ and $\Delta_i \stackrel{\text{def}}{=} (\underline{x}_i - \overline{x}_i)/2$ is the interval's half-width – do not intersect with each other [114].

The algorithms are as follows:

- In both algorithms, first, we sort all $2n$ endpoints of the narrowed intervals $\widetilde{x}_i - \frac{1+\alpha^2}{n} \cdot \Delta_i$ and $\widetilde{x}_i + \frac{1+\alpha^2}{n} \cdot \Delta_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$. This enables us to divide the real line into $2n+1$ zones $[x_{(i)}, x_{(i+1)}]$, where we denoted $x_{(0)} \stackrel{\text{def}}{=} -\infty$ and $x_{(2n+1)} \stackrel{\text{def}}{=} +\infty$.

- For each of zones $[x_{(i)}, x_{(i+1)}]$, we do the following: for each j from 1 to n, we pick the following value of x_j:

  - if $x_{(i+1)} < \widetilde{x}_j - \frac{1+\alpha^2}{n} \cdot \Delta_j$, then we pick $x_j = \overline{x}_j$;
  - if $x_{(i+1)} > \widetilde{x}_j + \frac{1+\alpha^2}{n} \cdot \Delta_j$, then we pick $x_j = \underline{x}_j$;
  - for all other j, we consider both possible values $x_j = \overline{x}_j$ and $x_j = \underline{x}_j$.

  As a result, we get one or several sequences of x_j for each zone.

- To compute $\overline{U}$, for each of the sequences x_j, we check whether, for the selected values $x_1, \ldots, x_n$, the value of $E - \alpha \cdot \sigma$ is indeed within the corresponding zone, and if it

is, compute the value $U = E + k_0 \cdot \sigma$. Finally, we return the largest of the computed values U as $\overline{U}$.

- To compute $\underline{L}$, for each of the sequences x_j, we check whether, for the selected values $x_1, \ldots, x_n$, the value of $E + \alpha \cdot \sigma$ is indeed within the corresponding zone, and if it is, compute the value $L = E - k_0 \cdot \sigma$. Finally, we return the smallest of the computed values L as $\underline{L}$.

Justification of the above-described algorithms. The algorithm of computing $\overline{U}$ can be proven to be correct: Since $1 + (1/k_0)^2 < n$, we can conclude that the maximum of the function U is attained when for every i, either $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$. For each i, we will consider both these cases.

If the maximum is attained for $x_i = \overline{x}_i$, this means, in particular, that if we keep all the other values x_j the same ($x'_j = x_j$) but replace x_i by $x'_i = \underline{x}_i = x_i - 2 \cdot \Delta_i$, then the value U will decrease. We will denote the values of E, U, etc., that correspond to $(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_n)$, by E', U', etc. In these terms, the desired inequality takes the form $U \geq U'$, where $U = E + k_0 \cdot \sigma$ and $U' = E' + k_0 \cdot \sigma'$. We can represent this inequality as $k_0 \cdot \sigma \geq (E' - E) + k_0 \cdot \sigma'$, hence either $(E' - E) + k_0 \cdot \sigma' \leq 0$, or $k_0^2 \cdot \sigma^2 \geq (E' - E)^2 + k_0^2 \cdot (\sigma')^2 + 2(E - E') \cdot k_0 \cdot \sigma'$. In the second case, we move the terms linear in σ' to one side of the inequality and square both sides again. As a result, we get an inequality that only contains variances $V = \sigma^2 = M - E^2$ (where M is the sample second moment) and $V' = (\sigma')^2 = M' - (E')^2$ and no longer contains square roots.

For our choice of x'_i, we have $E' = E - (2 \cdot \Delta_i)/n$ and

$$M' = M - \frac{4 \cdot \Delta_i \cdot x_i}{n} + \frac{4 \cdot \Delta_i^2}{n}.$$

Substituting these expressions into the above-described inequality and simplifying the resulting algebraic expression, we conclude that

$$\widetilde{x}_i + \Delta_i \cdot \frac{1 + \alpha^2}{n} \geq E - \alpha \cdot \sigma.$$

Similarly, if the maximum is attained for $x_i = \overline{x}_i$, this means, in particular, that if we keep all the other values x_j the same but replace x_i by $x_i' = \overline{x}_i = x_i + 2 \cdot \Delta_i$, then the value U will decrease. This property leads to the inequality

$$\widetilde{x}_i - \Delta_i \cdot \frac{1+\alpha^2}{n} \le E - \alpha \cdot \sigma.$$

So:

- if $x_i = \overline{x}_i$, then $E - \alpha \cdot \sigma \le \widetilde{x}_i + \Delta_i \cdot \frac{1+\alpha^2}{n}$;
- if $x_i = \underline{x}_i$, then $E - \alpha \cdot \sigma \ge \widetilde{x}_i - \Delta_i \cdot \frac{1+\alpha^2}{n}$.

Therefore, if we know the value of $E - \alpha \cdot \sigma$, then:

- if $\widetilde{x}_i + \Delta_i \cdot \frac{1+\alpha^2}{n} < E - \alpha \cdot \sigma$, then we cannot have $x_i = \overline{x}_i$ hence $x_i = \underline{x}_i$;
- similarly, if $\widetilde{x}_i - \Delta_i \cdot \frac{1+\alpha^2}{n} > E - \alpha \cdot \sigma$, then we cannot have $x_i = \underline{x}_i$ hence $x_i = \overline{x}_i$.

The only case when we do not know what value to choose is the case when

$$\widetilde{x}_i - \Delta_i \cdot \frac{1+\alpha^2}{n} \le E - \alpha \cdot \sigma \le \widetilde{x}_i + \Delta_i \cdot \frac{1+\alpha^2}{n},$$

i.e., when the value $E - \alpha \cdot \sigma$ belongs to the i-th narrowed interval; in this case, we can, in principle, have both $x_i = \underline{x}_i$ and $x_i = \overline{x}_i$. Thus, the algorithm is indeed correct.

This algorithm of computing $\overline{U}$ can be proven to take quadratic time. Indeed, once we know where E is with respect to the endpoints of all narrowed intervals, we can determine the values of all optimal x_i – except for those that are within this narrowed interval.

Since we consider the case when no narrowed intervals can have a common point, we have no more than 1 undecided values x_i. For each zone and for each of these combinations, we need a linear time ($O(n)$) to compute U. There are $O(n)$ zones, so the overall number of steps is $O(n^2)$.

For computing $\underline{L}$, the proof is, in effect, the same.

Comment. These algorithms also work when, for some fixed K, no more than K "narrowed" intervals $\left[\widetilde{x}_i - \Delta_i \cdot \frac{1+\alpha^2}{n}, \widetilde{x}_i + \Delta_i \cdot \frac{1+\alpha^2}{n}\right]$ can have a common point [114].

4.10.3 New Results

We use extend known efficient algorithms for computing $\overline{U}$ and $\underline{L}$ from the case of narrow intervals to a more general case when no two narrowed intervals $\left[\widetilde{x}_i - \Delta_i \cdot \dfrac{1+\alpha^2}{n}, \widetilde{x}_i + \Delta_i \cdot \dfrac{1+\alpha^2}{n}\right]$ are proper subsets of one another. This is a more general case because if they do not intersect, them, of course, they cannot be proper subsets of one another – in the sense that one of them is a subset of the interior of the second one. This new case is also a more general case than the case of single measuring instrument.

The new algorithm for computing $\overline{U}$ and $\underline{L}$. Let us first describe the algorithm itself; in the next section, we provide the justification for this algorithm.

- First, we sort of the values $\widetilde{x}_i$ into an increasing sequence. Without losing generality, we can assume that
$$\widetilde{x}_1 \le \widetilde{x}_2 \le \ldots \le \widetilde{x}_n.$$
- Then, for every k from 0 to n, we compute the value $V^{(k)} = M^{(k)} - (E^{(k)})^2$ of the population variance V for the vector $x^{(k)} = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$, and we compute $U^{(k)} = E^{(k)} + k_0 \cdot \sqrt{V^{(k)}}$.
- Finally, we compute $\overline{U}$ as the largest of $n+1$ values $U^{(0)}, \ldots, U^{(n)}$.

To compute the values $V^{(k)}$, first, we explicitly compute $M^{(0)}$, $E^{(0)}$, and $V^{(0)} = M^{(0)} - (E^{(0)})^2$. Once we know the values $M^{(k)}$ and $E^{(k)}$, we can compute
$$M^{(k+1)} = M^{(k)} + \frac{1}{n} \cdot (\underline{x}_{k+1})^2 - \frac{1}{n} \cdot (\overline{x}_{k+1})^2$$
and $E^{(k+1)} = E^{(k)} + \dfrac{1}{n} \cdot \underline{x}_{k+1} - \dfrac{1}{n} \cdot \overline{x}_{k+1}$.

Proof of the bound on the number of computation steps. It is well known that sorting takes $O(n \cdot \log(n))$ steps; see, e.g., [42]. Computing the initial values $M^{(0)}$, $E^{(0)}$, and $V^{(0)}$ takes linear time $O(n)$. For each k from 0 to $n-1$, we need a constant number of

steps to compute the next values $M^{(k+1)}$, $E^{(k+1)}$, and $V^{(k+1)}$. Computing $U^{(k+1)}$ also takes a constant number of steps. Finally, finding the largest of $n+1$ values $U^{(k)}$ also takes $O(n)$ steps. Thus, overall, we need

$$O(n \cdot \log(n)) + O(n) + O(n) + O(n) = O(n \cdot \log(n)) \text{ steps.}$$

It is worth mentioning that if the measurement results $\widetilde{x}_i$ are already sorted, then we only need linear time to compute $\overline{U}$.

Proof of the algorithm correctness. We have already mentioned that the maximum $\overline{U}$ of the function U is attained at a vector $x = (x_1, \ldots, x_n)$ in which each value x_i is equal either to $\underline{x}_i$ or to $\overline{x}_i$.

To justify our algorithm, we need to prove that this maximum is attained at one of the vectors $x^{(k)}$ in which all the lower bounds $\underline{x}_i$ precede all the upper bounds $\overline{x}_i$. We will prove this by reduction to a contradiction. Indeed, let us assume that the maximum is attained at a vector x in which one of the lower bounds follows one of the upper bounds. In each such vector, let i be the largest upper bound index followed by the lower bound; then, in the optimal vector x, we have $x_i = \overline{x}_i$ and $x_{i+1} = \underline{x}_{i+1}$.

Since the maximum is attained for $x_i = \overline{x}_i$, replacing it with $\underline{x}_i = \overline{x}_i - 2 \cdot \Delta_i$ will either decrease the value of U or keep it unchanged. Let us describe how U changes under this replacement. Since U is defined in terms of E, M, and V, let us first describe how E, M, and V change under this replacement. In the sum for M, we replace $(\overline{x}_i)^2$ with

$$(\underline{x}_i)^2 = (\overline{x}_i - 2 \cdot \Delta_i)^2 = (\overline{x}_i)^2 - 4 \cdot \Delta_i \cdot \overline{x}_i + 4 \cdot \Delta_i^2.$$

Thus, the value M changes into $M + \Delta M_i$, where

$$\Delta M_i = -\frac{4}{n} \cdot \Delta_i \cdot \overline{x}_i + \frac{4}{n} \cdot \Delta_i^2. \tag{4.10.1}$$

The population mean E changes into $E + \Delta E_i$, where

$$\Delta E_i = -\frac{2 \cdot \Delta_i}{n}. \tag{4.10.2}$$

Thus, the value E^2 changes into $(E + \Delta E_i)^2 = E^2 + \Delta(E^2)_i$, where

$$\Delta(E^2)_i = 2 \cdot E \cdot \Delta E_i + \Delta E_i^2 = -\frac{4}{n} \cdot E \cdot \Delta_i + \frac{4}{n^2} \cdot \Delta_i^2. \tag{4.10.3}$$

So, the variance V changes into $V + \Delta V_i$, where

$$\Delta V_i = \Delta M_i - \Delta(E^2)_i =$$

$$-\frac{4}{n} \cdot \Delta_i \cdot \overline{x}_i + \frac{4}{n} \cdot \Delta_i^2 + \frac{4}{n} \cdot E \cdot \Delta_i - \frac{4}{n^2} \cdot \Delta_i^2 =$$

$$\frac{4}{n} \cdot \Delta_i \cdot \left(-\overline{x}_i + \Delta_i + E - \frac{\Delta_i}{n} \right).$$

By definition, $\overline{x}_i = \widetilde{x}_i + \Delta_i$, hence $-\overline{x}_i + \Delta_i = -\widetilde{x}_i$. Thus, we conclude that

$$\Delta V_i = \frac{4}{n} \cdot \Delta_i \cdot \left(-\widetilde{x}_i + E - \frac{\Delta_i}{n} \right). \tag{4.10.4}$$

The function $U = E + k_0 \cdot \sigma$ attains its maximum if and only if the function $u \stackrel{\text{def}}{=} \alpha \cdot U = \alpha \cdot E + \sigma$ attains its maximum. After the change, the value u changes into

$$u + \Delta u_i = \alpha \cdot (E + \Delta E_i) + \sqrt{V + \Delta V_i},$$

so the condition $u + \Delta u_i \leq u$ leads to

$$\alpha \cdot (E + \Delta E_i) + \sqrt{V + \Delta V_i} \leq \alpha \cdot E + \sigma.$$

By moving the term proportional to α to the right-hand side, we conclude that $\sqrt{V + \Delta V_i} \leq \sigma - \alpha \cdot \Delta E_i$. In the new inequality, the left-hand side is the new value of the standard deviation, so it is a non-negative number, hence the right-hand side is also non-negative, so we can square both sides of the inequality and conclude that

$$V + \Delta V_i \leq \sigma^2 - 2 \cdot \alpha \cdot \sigma \cdot \Delta E_i + \alpha^2 \cdot (\Delta E_i)^2.$$

Moving all the terms to the left-hand side and using the fact that $V = \sigma^2$, we conclude that

$$z_i \stackrel{\text{def}}{=} \Delta V_i + 2 \cdot \alpha \cdot \sigma \cdot \Delta E_i - \alpha^2 \cdot (\Delta E_i)^2 \leq 0. \tag{4.10.5}$$

Substituting the known values of ΔV_i and ΔE_i, we get:

$$z_i = \frac{4}{n} \cdot \Delta_i \cdot e_i, \tag{4.10.6a}$$

where

$$e_i = -\widetilde{x}_i + E - \frac{\Delta_i}{n} - \alpha \cdot \sigma - \alpha^2 \cdot \frac{\Delta_i}{n},$$

i.e.,

$$e_i = (E - \alpha \cdot \sigma) - \left(\widetilde{x}_i + \frac{1+\alpha^2}{n} \cdot \Delta_i\right). \tag{4.10.6b}$$

Thus, from $z_i \le 0$, we conclude that

$$E - \alpha \cdot \sigma \le \widetilde{x}_i + \frac{1+\alpha^2}{n} \cdot \Delta_i. \tag{4.10.7}$$

Similarly, since the maximum of u is attained for $x_{i+1} = \underline{x}_{i+1}$, replacing it with $\overline{x}_{i+1} = \underline{x}_{i+1} + 2 \cdot \Delta_{i+1}$ will either decrease the value of u or keep it unchanged. Let us describe how variance changes under this replacement. In the sum for M, we replace $(\underline{x}_{i+1})^2$ with

$$(\overline{x}_{i+1})^2 = (\underline{x}_{i+1} + 2 \cdot \Delta_{i+1})^2 = (\underline{x}_{i+1})^2 + 4 \cdot \Delta_{i+1} \cdot \underline{x}_{i+1} + 4 \cdot \Delta_{i+1}^2.$$

Thus, the value M changes into $M + \Delta M_{i+1}$, where

$$\Delta M_{i+1} = \frac{4}{n} \cdot \Delta_{i+1} \cdot \underline{x}_{i+1} + \frac{4}{n} \cdot \Delta_{i+1}^2. \tag{4.10.8}$$

The population mean E changes into $E + \Delta E_{i+1}$, where

$$\Delta E_{i+1} = \frac{2 \cdot \Delta_{i+1}}{n}. \tag{4.10.9}$$

Thus, the value E^2 changes into

$$(E + \Delta E_{i+1})^2 = E^2 + \Delta(E^2)_{i+1},$$

where

$$\Delta(E^2)_{i+1} = 2 \cdot E \cdot \Delta E_{i+1} + \Delta E_{i+1}^2 = \frac{4}{n} \cdot E \cdot \Delta_{i+1} + \frac{4}{n^2} \cdot \Delta_{i+1}^2. \tag{4.10.10}$$

So, the variance V changes into $V + \Delta V_{i+1}$, where

$$\Delta V_{i+1} = \Delta M_{i+1} - \Delta(E^2)_{i+1} =$$

$$\frac{4}{n} \cdot \Delta_{i+1} \cdot \underline{x}_{i+1} + \frac{4}{n} \cdot \Delta_{i+1}^2 - \frac{4}{n} \cdot E \cdot \Delta_{i+1} - \frac{4}{n^2} \cdot \Delta_{i+1}^2 =$$

$$\frac{4}{n} \cdot \Delta_{i+1} \cdot \left(\underline{x}_{i+1} + \Delta_{i+1} - E - \frac{\Delta_{i+1}}{n} \right).$$

By definition, $\underline{x}_{i+1} = \widetilde{x}_{i+1} - \Delta_{i+1}$, hence $\underline{x}_{i+1} + \Delta_{i+1} = \widetilde{x}_{i+1}$. Thus, we conclude that

$$\Delta V_{i+1} = \frac{4}{n} \cdot \Delta_{i+1} \cdot \left(\widetilde{x}_{i+1} - E - \frac{\Delta_{i+1}}{n} \right). \tag{4.10.11}$$

Since u attains maximum at x, we have $\Delta u_{i+1} \le 0$, i.e., $z_{i+1} \le 0$, where

$$z_{i+1} \stackrel{\text{def}}{=} \Delta V_{i+1} + 2 \cdot \alpha \cdot \sigma \cdot \Delta E_{i+1} - \alpha^2 \cdot (\Delta E_{i+1})^2. \tag{4.10.12}$$

Substituting the expressions (4.10.11) for ΔV_{i+1} and (4.10.9) for ΔE_{i+1} into this formula, we conclude that

$$z_{i+1} = \frac{4}{n} \cdot \Delta_{i+1} \cdot e_{i+1}, \tag{4.10.13a}$$

where

$$e_{i+1} \stackrel{\text{def}}{=} -(E - \alpha \cdot \sigma) + \left(\widetilde{x}_{i+1} - \frac{1 + \alpha^2}{n} \cdot \Delta_{i+1} \right) \tag{4.10.13b}$$

and

$$E - \alpha \cdot \sigma \ge \widetilde{x}_{i+1} - \frac{1 + \alpha^2}{n} \cdot \Delta_{i+1}. \tag{4.10.14}$$

We can also change *both* x_i and x_{i+1} at the same time. In this case, from the fact that u attains the maximum at x, we conclude that $u + \Delta u \le u$, i.e., that

$$z \stackrel{\text{def}}{=} \Delta V + 2 \cdot \alpha \cdot \sigma \cdot \Delta E - \alpha^2 \cdot (\Delta E)^2. \tag{4.10.15}$$

Here, the change ΔM in M is simply the sum of the changes coming from x_i and x_{i+1}:

$$\Delta M = \Delta M_i + \Delta M_{i+1}, \tag{4.10.16}$$

and the change ΔE in E is also the sum of the corresponding changes:

$$\Delta E = \Delta E_i + \Delta E_{i+1}. \tag{4.10.17}$$

So, for

$$\Delta V = \Delta M - \Delta(E^2) = \Delta M - 2 \cdot E \cdot \Delta E - \Delta E^2,$$

we get

$$\Delta V = \Delta M_i + \Delta M_{i+1} - 2 \cdot E \cdot \Delta E_i - 2 \cdot E \cdot \Delta E_{i+1} -$$

$$(\Delta E_i)^2 - (\Delta E_{i+1})^2 - 2 \cdot \Delta E_i \cdot \Delta E_{i+1}.$$

Hence,

$$\Delta V = (\Delta M_i - 2 \cdot E \cdot \Delta E_i - (\Delta E_i)^2) +$$

$$(\Delta M_{i+1} - 2 \cdot E \cdot \Delta E_{i+1} - (\Delta E_{i+1})^2) - 2 \cdot \Delta E_i \cdot \Delta E_{i+1},$$

i.e.,

$$\Delta V = \Delta V_i + \Delta V_{i+1} - 2 \cdot \Delta E_i \cdot \Delta E_{i+1}. \tag{4.10.18}$$

Substituting expressions (4.10.16), (4.10.17), and (4.10.18) into the formula (4.10.15) for z, we conclude that

$$z = \Delta V + 2 \cdot \alpha \cdot \sigma \cdot \Delta E - \alpha^2 \cdot (\Delta E)^2 =$$

$$\Delta V_i + \Delta V_{i+1} - 2 \cdot \Delta E_i \cdot \Delta E_{i+1} + 2\alpha \cdot \sigma \cdot \Delta E_i + 2\alpha \cdot \sigma \cdot \Delta E_{i+1} -$$

$$\alpha^2 \cdot (\Delta E_i)^2 - \alpha^2 \cdot (\Delta E_{i+1})^2 - 2 \cdot \alpha^2 \cdot \Delta E_i \cdot \Delta E_{i+1}.$$

Hence,

$$z = (\Delta V_i + 2 \cdot \alpha \cdot \sigma \cdot \Delta E_i - \alpha^2 \cdot (\Delta E_i)^2) +$$

$$(\Delta V_{i+1} + 2 \cdot \alpha \cdot \sigma \cdot \Delta E_{i+1} - \alpha^2 \cdot (\Delta E_{i+1})^2) - 2 \cdot (1 + \alpha^2) \cdot \Delta E_i \cdot \Delta E_{i+1}.$$

From the formulas (4.10.5) and (4.10.12), we know that the first expression is z_i and that the second expression is z_{i+1}, so

$$z = z_i + z_{i+1} - 2 \cdot (1 + \alpha^2) \cdot \Delta E_i \cdot \Delta E_{i+1}.$$

We already have the expressions (4.10.6), (4.10.13), (4.10.2), and (4.10.9) for, correspondingly, z_i, z_{i+1}, ΔE_i, and ΔE_{i+1}, so we conclude that $z = \dfrac{4}{n} \cdot D(E')$, where $E' \stackrel{\text{def}}{=} E - \alpha \cdot \sigma$ and

$$D(E') \stackrel{\text{def}}{=} \Delta_i \cdot \left(E' - \left(\widetilde{x}_i + \frac{1 + \alpha^2}{n} \cdot \Delta_i \right) \right) +$$

$$\Delta_{i+1} \cdot \left(-E' + \left(\widetilde{x}_{i+1} - \frac{1+\alpha^2}{n} \cdot \Delta_{i+1}\right)\right) + 2 \cdot (1+\alpha^2) \cdot \frac{\Delta_i \cdot \Delta_{i+1}}{n}. \tag{4.10.19}$$

Since $z \le 0$, we have $D(E') \le 0$ (for the value $E' = E - \alpha \cdot \sigma$ corresponding to the optimizing vector x).

The expression $D(E')$ is a linear function of E'. From (4.10.7) and (4.10.14), we know that

$$\widetilde{x}_{i+1} - \frac{1+\alpha^2}{n} \cdot \Delta_{i+1} \le E' \le \widetilde{x}_i + \frac{1+\alpha^2}{n} \cdot \Delta_i.$$

For $E' = E^- \stackrel{\text{def}}{=} \widetilde{x}_{i+1} - \dfrac{1+\alpha^2}{n} \cdot \Delta_{i+1}$, we have

$$D(E^-) = \Delta_i \cdot f_i + \frac{2 \cdot (1+\alpha^2)}{n} \cdot \Delta_i \cdot \Delta_{i+1},$$

where

$$f_i \stackrel{\text{def}}{=} -\widetilde{x}_i + \widetilde{x}_{i+1} - \frac{1+\alpha^2}{n} \cdot \Delta_{i+1} - \frac{1+\alpha^2}{n} \cdot \Delta_i,$$

hence $D(E^-) = \Delta_i \cdot g_i$, where

$$g_i \stackrel{\text{def}}{=} -\widetilde{x}_i + \widetilde{x}_{i+1} + \frac{1+\alpha^2}{n} \cdot \Delta_{i+1} - \frac{1+\alpha^2}{n} \cdot \Delta_i.$$

We assumed that no narrowed interval is a proper subset of any other. How can we describe this condition in algebraic terms? Let us denote $\delta_i \stackrel{\text{def}}{=} \dfrac{1+\alpha^2}{n} \cdot \Delta_i$; then, the i-th narrowed interval has the form $[\widetilde{x}_i - \delta_i, \widetilde{x}_i + \delta_i]$. If $[\widetilde{x}_i - \delta_i, \widetilde{x}_i + \delta_i]$ is a proper subinterval of $[\widetilde{x}_j - \delta_j, \widetilde{x}_j + \delta_j]$, this means that $\widetilde{x}_i - \delta_i > \widetilde{x}_j - \delta_j$ and $\widetilde{x}_i + \delta_i < \widetilde{x}_j + \delta_j$, i.e., equivalently, that

$$\delta_i - \delta_j < \widetilde{x}_i - \widetilde{x}_j < \delta_j - \delta_i.$$

This inequality is equivalent to $\delta_j > \delta_i$ and $|\widetilde{x}_i - \widetilde{x}_j| < \delta_j - \delta_i$. Similarly, the condition that the j-th narrowed interval is a proper subinterval of the i-th is equivalent to $\delta_j < \delta_i$ and $|\widetilde{x}_i - \widetilde{x}_j| < \delta_i - \delta_j$. Both cases can be described by a single inequality $|\widetilde{x}_i - \widetilde{x}_j| < |\delta_i - \delta_j|$. Thus, the condition that no narrowed interval can be a proper subinterval of any other narrowed interval can be described as

$$|\widetilde{x}_i - \widetilde{x}_j| \ge |\delta_i - \delta_j|. \tag{4.10.20}$$

In particular, we have $|\widetilde{x}_i - \widetilde{x}_{i+1}| \geq |\delta_i - \delta_{i+1}|$.

Let us first consider the case when

$$|\widetilde{x}_{i+1} - x_i| > |\delta_i - \delta_{i+1}|.$$

Since the values $\widetilde{x}_i$ are sorted in increasing order, we have $\widetilde{x}_{i+1} \geq \widetilde{x}_i$, hence

$$\widetilde{x}_{i+1} - \widetilde{x}_i = |\widetilde{x}_{i+1} - \widetilde{x}_i| > |\delta_i - \delta_{i+1}| \geq \delta_i - \delta_{i+1}.$$

So, we conclude that $D(E^-) > 0$.

For $E = E^+ \stackrel{\text{def}}{=} \widetilde{x}_i + \dfrac{1+\alpha^2}{n} \cdot \Delta_i$, we have

$$D(E^+) = \Delta_{i+1} \cdot f_{i+1} + \frac{2 \cdot (1+\alpha^2)}{n} \cdot \Delta_i \cdot \Delta_{i+1},$$

where

$$f_{i+1} \stackrel{\text{def}}{=} -\widetilde{x}_i + \widetilde{x}_{i+1} - \frac{1+\alpha^2}{n} \cdot \Delta_{i+1} - \frac{1+\alpha^2}{n} \cdot \Delta_i,$$

hence $D(E^+) = \Delta_{i+1} \cdot g_{i+1}$, where

$$g_{i+1} \stackrel{\text{def}}{=} -\widetilde{x}_i + \widetilde{x}_{i+1} + \frac{1+\alpha^2}{n} \cdot \Delta_i - \frac{1+\alpha^2}{n} \cdot \Delta_{i+1}.$$

Here, from $|\widetilde{x}_{i+1} - \widetilde{x}_i| > |\delta_i - \delta_{i+1}|$, we also conclude that $D(E^+) > 0$.

Since the linear function $D(E')$ is positive on both endpoints of the interval $[E^-, E^+]$, it must be positive for every value E' from this interval, which contradicts to our conclusion that $D(E') \leq 0$ for the actual value $E' = E - \alpha \cdot \sigma \in [E^-, E^+]$. This contradiction shows that the maximum of U is indeed attained at one of the values $x^{(k)}$, hence the algorithm is justified.

The general case when $|\widetilde{x}_i - \widetilde{x}_j| \geq |\delta_i - \delta_j|$ can be obtained as a limit of cases when we have strict inequality. Since the function U is continuous, the value $\overline{U}$ continuously depends on the input bounds, so by tending to a limit, we can conclude that our algorithm works in the general case as well.

Comment. This algorithm can be naturally extended to an $O(n^m)$ time algorithm for the case of m measuring instruments.

4.11 Other Statistical Characteristics – Skewness

4.11.1 Formulation of the Problem

Once we have several results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of measuring some physical quantity, to check whether the distribution is symmetric, we compute its sample skewness $S(\widetilde{x}_1, \ldots, \widetilde{x}_n) = \frac{1}{n}\sum_{i=1}^{n}(\widetilde{x}_i - E)^3$.

By the *interval skewness* $\mathbf{S} = [\underline{S}, \overline{S}]$ of the interval data, we mean the interval

$$\mathbf{S} \stackrel{\text{def}}{=} \{S(x_1, \ldots, x_n) \,|\, x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}$$

filled by the values $S(x_1, \ldots, x_n)$ corresponding to different $x_i \in \mathbf{x}_i$.

For computing the range of skewness under interval uncertainty, no effective algorithm was previously known.

4.11.2 New Results

Reduction to computing $\overline{S}$. In order to compute the interval $\mathbf{S}$, we must compute both endpoints $\underline{S}$ and $\overline{S}$ of this interval.

Let us first show that if we can compute $\overline{S}$, then we can easily compute $\underline{S}$ as well. Indeed, skewness is an odd function: $S(-x_1, \ldots, -x_n) = -S(x_1, \ldots, x_n)$; thus, for intervals $-\mathbf{x}_i = -[\underline{x}_i, \overline{x}_i] = [-\overline{x}_i, -\underline{x}_i]$, we have $\mathbf{S}(-\mathbf{x}_1, \ldots, -\mathbf{x}_n) = -\mathbf{S}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$. From this relation between the skewness intervals, we can conclude that $\overline{S}(-\mathbf{x}_1, \ldots, -\mathbf{x}_n) = -\underline{S}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

Thus, if we know how to compute $\overline{S}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ for an arbitrary collection of intervals $\mathbf{x}_i$, we can thus compute $\underline{S}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ as $-\overline{S}(-\mathbf{x}_1, \ldots, -\mathbf{x}_n)$

In view of this comment, in the remaining part of the section, we will only consider an algorithm for computing $\overline{S}$.

The quadratic time algorithm for computing $\overline{S}$ in case of subset property.

1°. Since the interval data satisfies the subset property, after we sort these elements in lexicographic order, it can be shown that both the lower endpoints $\underline{x}_i$ and the upper endpoints $\overline{x}_i$ are sorted in non-decreasing order: $\underline{x}_i \le \underline{x}_{i+1}$ and $\overline{x}_i \le \overline{x}_{i+1}$.

2°. The maximum of a differentiable function $S(x_1, \ldots, x_n)$ on an interval $[\underline{x}_i, \overline{x}_i]$ can be attained either in an internal point of this interval, or at one of the endpoints.

If the maximum is attained at an internal point, then the first derivative is 0 $\left(\dfrac{\partial S}{\partial x_i} = 0\right)$ and the second derivative should be non-positive $\left(\dfrac{\partial^2 S}{\partial x_i^2} \le 0\right)$.

If the maximum is attained at the left endpoint, the function S cannot be increasing at this point, so we must have $\dfrac{\partial S}{\partial x_i} \le 0$. Similarly, if the maximum is attained at the right endpoint, the function S cannot be decreasing at this point, so we must have $\dfrac{\partial S}{\partial x_i} \ge 0$.

For skewness,

$$\frac{\partial S}{\partial x_i} = \frac{3}{n} \cdot (x_i - E)^2 - \frac{3}{n} \cdot \sum_{j=1}^{n} (x_j - E)^2 \cdot \frac{\partial E}{\partial x_i}.$$

Since $\dfrac{\partial E}{\partial x_i} = \dfrac{1}{n}$, we thus get $\dfrac{\partial S}{\partial x_i} = \dfrac{3}{n} \cdot ((x_i - E)^2 - V)$. So, the first derivative of S has the same sign as the expression $(x_i - E)^2 - V$.

To compute the second derivative of S, we must take into account that $\dfrac{\partial V}{\partial x_i} = \dfrac{2}{n} \cdot (x_i - E)$, hence

$$\frac{\partial^2 S}{\partial x_i^2} = \frac{3}{n} \cdot \left(2(x_i - E) - 2(x_i - E) \cdot \frac{1}{n} - \frac{2}{n} \cdot (x_i - E) + \frac{2}{n} \cdot \sum_{j=1}^{n} (x_j - E) \cdot \frac{1}{n}\right).$$

Since $\sum\limits_{j=1}^{n} (x_j - E) = 0$, we conclude that

$$\frac{\partial^2 S}{\partial x_i^2} = \frac{3}{n} \cdot 2 \cdot \left(1 - \frac{2}{n}\right) \cdot (x_i - E).$$

We have already mentioned that the problem of computing skewness only makes sense for $n > 2$, because for $n \le 2$, the skewness is identically 0. For $n > 2$, the second derivative $\dfrac{\partial^2 S}{\partial x_i^2}$ has the same sign as the expression $x_i - E$.

Thus, for skewness, we value x_i at which the maximum is attained satisfies one of the following three conditions:

- either $\underline{x}_i < x_i < \overline{x}_i$, $(x_i - E)^2 - V = 0$, and $x_i - E \le 0$,
- or $x_i = \underline{x}_i$ and $(x_i - E)^2 - V \le 0$,
- or $x_i = \overline{x}_i$ and $(x_i - E)^2 - V \ge 0$.

In the first case, $(x_i - E)^2 = V = \sigma^2$, hence $x_i - E = \pm\sigma$. Since $x_i - E \le 0$, we cannot have $x_i - E = \sigma$, so in this case, $x_i = E - \sigma$. In the second case, $(x_i - E)^2 \le V = \sigma^2$, hence $E - \sigma \le x_i \le E + \sigma$. In the third case, $(x_i - E)^2 \ge V = \sigma^2$, so either $x_i \le E - \sigma$ or $x_i \ge E + \sigma$. So:

- either $\underline{x}_i < x_i = E - \sigma < \overline{x}_i$,
- or $x_i = \underline{x}_i$ and $E - \sigma \le \underline{x}_i \le E + \sigma$,
- or $x_i = \overline{x}_i$ and either $\overline{x}_i \le E - \sigma$ or $\overline{x}_i \ge E + \sigma$.

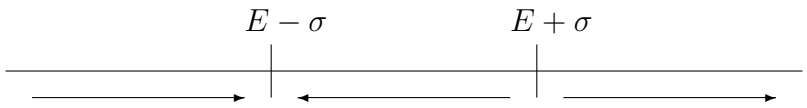
In all three cases, the desired maximum of the skewness S is attained when x_i is either at one of the endpoints of the corresponding interval $\mathbf{x}_i$, or has the value $\mu \stackrel{\text{def}}{=} E - \sigma$.

3°. Let us now deduce a more specific information about the values x_i at which the maximum is attained.

Based on the above description of possible cases, once we know how the intervals are located in relation to $E - \sigma$ and $E + \sigma$, we can sometimes uniquely determine the value x_i at which the maximum is attained. Namely,

- If $\overline{x}_i \le E - \sigma$, then the maximum cannot be attained at an internal point and it cannot be attained at the value $\underline{x}_i$, so it is attained when $x_i = \overline{x}_i$.
- If $\underline{x}_i \le E - \sigma \le \overline{x}_i \le E + \sigma$, then the maximum can only be attained when $x_i = E - \sigma$.
- If $E - \sigma \le \underline{x}_i \le E + \sigma$, then the maximum is attained at $x_i = \underline{x}_i$.
- Finally, if $E + \sigma \le \underline{x}_i$, then the maximum is attained at $x_i = \overline{x}_i$.

These conclusions can be described in the following graphical manner, in which the arrows indicate the direction towards the corresponding maximum:



The only case when we cannot exactly determine the optimal value x_i is when the interval $\mathbf{x}_i$ contains the value $E+\sigma$: it this case, we may have $x_i = \overline{x}_i$, and we may also have $x_i = \max(E-\sigma, \underline{x}_i)$.

4°. Let us show that the maximum of skewness is always attained at a vector $x = (x_1, \ldots, x_n)$ which can be divided into three consequent fragments (some of which may be empty):

- first, we have values $\overline{x}_i$ which are smaller than $E-\sigma$;
- then, we have the values $\max(E-\sigma, \underline{x}_i)$;
- finally, we have the values $\overline{x}_i$ which are larger than $E+\sigma$.

All the intervals $\mathbf{x}_i$ that do not contain $E+\sigma$ inside naturally fall into this scheme. The only intervals that we do need to consider to prove this result are the intervals that do contain $E+\sigma$. For each of these intervals, the corresponding values x_i are either $\max(E-\sigma, \underline{x}_i)$ or $\overline{x}_i$. What we claim is that after we sort the intervals in lexicographic order, we will first have the values equal to $\max(E-\sigma, \underline{x}_i)$, and then the values equal to $\overline{x}_i$. In other words, once we have a value $x_i = \overline{x}_i$, all the following values will also be of the same type.

We will show that if there is an optimizing vector at which this condition is not satisfied, then we can rearrange it into a new vector with the same optimal value of S for which this condition holds.

Indeed, let us start with a vector for which, for some i, for two consequent intervals $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, a value $x_i = \overline{x}_i \geq E + \sigma$ is followed by a value $x_{i+1} = \max(E - \sigma, \underline{x}_{i+1}) \leq E + \sigma$. If there are several such indices i, we take the smallest i with this property.

According to Part 2 of this proof, we have $\underline{x}_i \leq \underline{x}_{i+1} \leq E + \sigma$ and $E + \sigma \leq \overline{x}_i \leq \overline{x}_{i+1}$. Thus, $x_i = \overline{x}_i \leq \overline{x}_{i+1}$ and $x_i = \overline{x}_i \geq E + \sigma \geq \underline{x}_{i+1}$; hence, $x_i \in \mathbf{x}_{i+1}$. Similarly, $x_{i+1} \in \mathbf{x}_i$. Thus, we can "swap" the values x_i and x_{i+1}: as a new value of x_i, we take the old value of x_{i+1}, and vice versa. The swap does not change the average E and does not change the sample skewness S, so the function S attains the maximum at the new values as well.

As a result of this swap, if there is now a value i' for which $\overline{x}_{i'}$ is followed by $\max(E - \sigma, \underline{x}_{i'+1})$, this value i' has to be equal to at least $i + 1$. If there still is such an index i', we apply a new swap again and thus again increase the smallest problematic value i. After $\leq n$ such swaps, there will be no problematic cases anymore, so we will get a sequence which has the desired property.

5°. To determine the optimal vector x, we must thus select a zone $[x_{(p)}, x_{(p+1)}]$ that contains $\mu = E - \sigma$, and an index k at which the optimal value x_i switches from $\max(\mu, \underline{x}_i)$ to $\overline{x}_i$.

Once p and k are fixed, we can uniquely determine each of the optimal values x_i – some as known numbers, some as equal to the (unknown) value μ:

- when $\overline{x}_i \leq x_{(p)}$, we have $x_i = \overline{x}_i$;
- when $\underline{x}_i < x_{(p)} < x_{(p+1)} \leq \overline{x}_i$ and $i < k$, we have $x_i = \mu$;
- when $x_{(p+1)} \leq \underline{x}_i$ and $i < k$, we have $x_i = \underline{x}_i$;
- finally, when $i \geq k$, we have $x_i = \overline{x}_i$.

To find μ, we must use the fact that $\mu = E - \sigma$. Specifically, the average E can be determined as

$$\frac{1}{n} \cdot \sum_{i \in N'} x_i + \frac{n - n'}{n} \cdot (E - \sigma) = E,$$

where the sum is taken over the set N' of all the indices for which x_i is known, and n' is the total number of such indices. Similarly, the sample second moment $E^2 + \sigma^2$ can be

determined as

$$\frac{1}{n}\cdot\sum_{i\in N'}x_i^2+\frac{n-n'}{n}\cdot(E-\sigma)^2=E^2+\sigma^2.$$

From the first of these equations, we can determine σ as a linear function of E. Substituting this expression into the second equation, we get a quadratic equation with the only unknown σ, from which we can determine σ. Then, we can use the first equation to find E – and hence find $\mu=E-\sigma$.

If the resulting value of μ is indeed within the zone $[x_{(p)},x_{(p+1)}]$, then we compute the sample skewness for the corresponding values x_i. Specifically, the skewness can be computed as

$$\frac{1}{n}\cdot\sum_{i=1}^{n}(x_i-E)^3=\frac{1}{n}\cdot\sum_{i=1}^{n}x_i^3-\frac{3\cdot E}{n}\cdot\sum_{i=1}^{n}x_i^2+\frac{3\cdot E^2}{n}\cdot\sum_{i=1}^{n}x_i-E^3=$$

$$\frac{1}{n}\cdot\sum_{i=1}^{n}x_i^3-\frac{3\cdot E}{n}\cdot\sum_{i=1}^{n}x_i^2+2\cdot E^3=$$

$$\frac{1}{n}\cdot\sum_{i\in N'}x_i^3+\frac{n-n'}{n}\cdot\mu^3-\frac{3\cdot E}{n}\cdot\left(\sum_{i\in N'}x_i^2+(n-n')\cdot\mu^2\right)+2\cdot E^3.$$

The largest of these skewnesses is the desired value $\overline{S}$.

6°. How much times does this algorithm take? Sorting takes time $O(n\cdot\log(n))$.

For n interval data points, we have $2n$ possible zone and n possible indices k – totally, $O(n^2)$ possible pairs (p,k). For the first pair, computing the corresponding values n', $\sum\limits_{i\in N'}x_i$, $\sum\limits_{i\in N'}x_i^2$, and $\sum\limits_{i\in N'}x_i^3$ takes linear time. For each next pair, we, in general, change one value in comparison with the previous pair, so each new computation takes a constant number of steps. Thus, for $O(n^2)$ pairs, we need $O(n^2)$ time. (In some cases, we change more than one value, but still, each value changes only once, so we still need $O(n^2)$ times.)

So, overall, we need time $O(n\cdot\log(n))+O(n)+O(n^2)=O(n^2)$.

Feasible algorithm for computing $\overline{S}$ in case of several MI. In this case, the interval data consists of m families of intervals such that within each family, no two intervals are proper subsets of each other.

For $\overline{S}$, we need to consider n^2 options for each of the m subsequences corresponding to a single MI; thus, overall, we must consider $(O(n^2))^m = O(n^{2m})$ possible combinations – hence we need time $O(n^{2m})$.

Chapter 5

Applications

In the previous chapter, we described algorithms for computing statistics under interval uncertainty. In this chapter, we describe applications of these algorithms and other similar algorithms. Specifically, we describe several application areas in which there is a need to take into account interval uncertainty in statistical data processing: the seismic inverse problem in geophysics, the problem of estimating and decreasing the clock cycle in computer chips, the problem of separating the core from the fragments in radar data processing, and the problem of inverse half-toning in image processing. For each of these applications, we describe the corresponding problem, explain why interval uncertainty is important, and describe algorithms for taking this interval uncertainty into consideration.

5.1 Geosciences

In many real-life situations, we have several types of uncertainty: measurement uncertainty can lead to probabilistic and/or interval uncertainty, expert estimates come with interval and/or fuzzy uncertainty, etc. In many situations, in addition to measurement uncertainty, we have prior knowledge coming from prior data processing, prior knowledge coming from prior interval constraints. In this dissertation, on the example of the seismic inverse problem, we show how to combine these different types of uncertainty.

5.1.1 Seismic Inverse Problem: A Brief Description

In evaluations of natural resources and in the search for natural resources, it is very important to determine earth structure. Our civilization greatly depends on

the things we extract from the Earth, such as fossil fuels (oil, coal, natural gas), minerals, and water. Our need for these commodities is constantly growing, and because of this growth, they are being exhausted. Even under the best conservation policies, there is (and there will be) a constant need to find new sources of minerals, fuels, and water.

The only sure-proof way to guarantee that there are resources such as minerals at a certain location is to actually drill a borehole and analyze the materials extracted. However, exploration for natural resources using indirect means began in earnest during the first half of the 20th century. The result was the discovery of many large relatively easy to locate resources such as the oil in the Middle East.

However, nowadays, most easy-to-access mineral resources have already been discovered. For example, new oil fields are mainly discovered either at large depths, or under water, or in very remote areas – in short, in the areas where drilling is very expensive. It is therefore desirable to predict the presence of resources as accurately as possible before we invest in drilling.

From previous exploration experiences, we usually have a good idea of what type of structures are symptomatic for a particular region. For example, oil and gas tend to concentrate near the top of natural underground domal structures. So, to be able to distinguish between more promising and less promising locations, it is desirable to determine the structure of the Earth at these locations. To be more precise, we want to know the structure at different depths z at different locations (x, y).

Data that we can use to determine the earth structure. In general, to determine the Earth structure, we can use different measurement results that can be obtained without actually drilling the boreholes: e.g., gravity and magnetic measurements, analyzing the travel-times and paths of seismic ways as they propagate through the earth, etc.

To get a better understanding of the Earth structure, we must rely on *active* seismic data – in other words, we must make artificial explosions, place sensors around them, and measure how the resulting seismic waves propagate. The most important information about

the seismic wave is the *travel-time* t_i, i.e., the time that it takes for the wave to travel from its source to the sensor. to determine the geophysical structure of a region, we measure seismic travel times and reconstruct velocities at different depths from these data. The problem of reconstructing this structure is called the *seismic inverse problem*.

Known algorithms for solving the seismic inverse problem: description, successes, limitations. We want to find the values of the velocity $v(\vec{x})$ at different 3-D points $\vec{x}$. Based on the finite number of measurements, we can only reconstruct a finite number of parameters. So, we take a rectangular grid and to reconstruct the velocities v_j at different grid points.

Algorithm for the forward problem: brief description. Once we know the velocities v_j in each cell j, we can then determine the paths which seismic waves take. Seismic waves travel along the shortest path – shortest in terms of time. It can be easily determined that for such paths, within each cell, the path is a straight line, and on the border between the two cells with velocities v and v', the direction of the path changes in accordance with Snell's law $\dfrac{\sin(\varphi)}{v} = \dfrac{\sin(\varphi')}{v'}$, where φ and φ' are the angles between the paths and the line orthogonal to the border between the cells. (If this formula requires $\sin(\varphi') > 1$, this means that this wave cannot penetrate into the neighboring cell at all; instead, it bounces back into the original cell with the same angle φ.)

In particular, we can thus determine the paths from the source to each sensor. The travel-time t_i along i-th path can then be determined as the sum of travel-times in different cells j through which this path passes: $t_i = \sum\limits_j \dfrac{\ell_{ij}}{v_j}$, where ℓ_{ij} denotes the length of the part of i-th path within cell j.

This formula becomes linear if we replace the original unknowns – velocities v_j – by their inverses $s_j \stackrel{\text{def}}{=} \dfrac{1}{v_j}$, called *slownesses*. In terms of slownesses, the formula for the travel-time takes the simpler form $t_i = \sum\limits_j \ell_{ij} \cdot s_j$.

Algorithm for the inverse problem: general description. There are several algorithms for solving this inverse problem; see, e.g., [83, 171, 214]. The most widely used is the following iterative algorithm proposed by John Hole [83].

At each stage of this algorithm, we have some approximation to the desired slownesses. We start with some reasonable initial slownesses, and we hope that after several iterations, we will be able to get slownesses which are much closer to the actual values.

At each iteration, we first use the currently known slownesses s_j to find the corresponding paths from the source to each sensor. Based on these paths, we compute the predicted values $t_i = \sum\limits_j \ell_{ij} \cdot s_j$ of travel-times.

Since the currently known slownesses s_j are only approximately correct, the travel-times t_i (which are predicted based on these slownesses) are approximately equal to the measured travel-times $\widetilde{t}_i$; there is, in general, a discrepancy $\Delta t_i \stackrel{\text{def}}{=} \widetilde{t}_i - t_i \neq 0$. It is therefore necessary to use these discrepancies to update the current values of slownesses, i.e., replace the current values s_j with corrected values $s_j + \Delta s_j$. The objective of this correction is eliminate (or at least decrease) the discrepancies $\Delta t_i \neq 0$. In other words, the objective is to make sure that for the corrected values of the slowness, the predicted travel-times are closer to $\widetilde{t}_i$.

Of course, once we have changed the slownesses, the shortest paths will also change; however, if the current values of slownesses are reasonable, the differences in slowness are not large, and thus, paths will not change much. Thus, in the first approximation, we can assume that the paths are the same, i.e., that for each i and j, the length ℓ_{ij} remains the same. In this approximation, the new travel-times are equal to $\sum \ell_{ij} \cdot (s_j + \Delta s_j)$. The desired condition is then $\sum \ell_{ij} \cdot (s_j + \Delta s_j) = \widetilde{t}_i$. Subtracting the formula $t_i = \sum\limits_j \ell_{ij} \cdot s_j$ from this expression, we conclude that the corrections Δs_j must satisfy the following system of (approximate) linear equations: $\sum \ell_{ij} \cdot \Delta s_j \approx \Delta t_i$.

Solving this system of linear equations is not an easy task, because we have many observations and many cell values and thus, many unknowns, and for a system of linear equations, computation time to solve it grows as a cube n^3 of the number of variables n.

So, instead of the standard methods for solving a system of linear equations, researchers use special faster geophysics-motivated techniques (described below) for solving the corresponding systems. These methods are described, in detail, in the next subsection.

Once we solve the corresponding system of linear equations, we compute the updated values Δs_j, compute the new (corrected) slownesses $s_j + \Delta s_j$, and repeat the procedure again. We stop when the discrepancies become small; usually, we stop when the mean square error $\frac{1}{n}\sum_{i=1}^{n}(\Delta t_i)^2$ no longer exceeds a given threshold. This threshold is normally set up to be equal to the measurement noise level, so that we stop iterations when the discrepancy between the model and the observations falls below the noise level – i.e., when, for all practical purposes, the model is adequate.

Algorithm for the inverse problem: details. Let us describe, in more detail, how the corresponding linear system of equations is usually solved. In other words, for a given cell j, how do we find the correction Δs_j to the current value of slowness s_j in this cell?

Let us first consider the simplified case when there is only path, and this path is going through the j-th cell. In this case, cells through which this path does not go does not need any correction. To find the corrections Δs_j for all the cells j through which this path goes, we only have one equation $\sum\limits_j \ell_{ij} \cdot \Delta s_j = \Delta t_i$. The resulting system of linear equations is clearly under-determined: we have a single equation to find the values of several variables Δs_j. Since the system is under-determined, we have a infinite number of possible solutions. Our objective is to select the most geophysical reasonable of these solutions.

For that, we can use the following idea. Our single observation involves several cells; we cannot distinguish between the effects of slownesses in different cells, we only observe the overall effect. Therefore, there is no reason to assume that the value Δs_j in one of these cells is different from the values in other cells. It is thus reasonable to assume that all these values are close to each other: $\Delta s_j \approx \Delta s_{j'}$. The least squares method enables us to describe this assumption as minimization of the objective function $\sum\limits_{j,j'}(\Delta s_j - \Delta s_{j'})^2$ under the condition that $\sum \ell_{ij} \cdot \Delta s_j = \Delta t_i$. The minimum is attained when all the values Δs_j are

equal. Substituting these equal values into the equation $\sum\limits_j \ell_{ij} \cdot \Delta s_j = \Delta t_i$, we conclude that $L_i \cdot \Delta s = \Delta t_i$, where $L_i = \sum\limits_j \ell_{ij}$ is the overall length of i-th path. Thus, in the simplified case in which there is only one path, to the slowness of each cell j along this path, we add the same value $\Delta s_j = \dfrac{\Delta t_i}{L_i}$.

Let us now consider the realistic case in which there are many paths, and moreover, for many cells j, there are many paths i which go through the corresponding cell. For a given cell j, based on each path i passing through this cell, we can estimate the correction Δs_j by the corresponding value $\Delta s_{ij} \stackrel{\text{def}}{=} \dfrac{\Delta t_i}{L_i}$. Since there are usually several paths going through the j-th cell, we have, in general, several different estimates $\Delta s_j \approx \Delta s_{ij}$. Again, the least squares approach leads to $\sum\limits_i (\Delta s_j - \Delta s_{ij})^2 \to \min$, hence to Δs_j as the arithmetic average of the values Δs_{ij}.

Comment. To take into account that paths with larger ℓ_{ij} provide more information, researchers also used weighted average, with weight increasing with ℓ_{ij}.

Successes of the known algorithms. The known algorithms have been actively used to reconstruct the slownesses, and, in many practical situations, they have led to reasonable geophysical models.

Limitations of the known algorithms. Often, the velocity model that is returned by the existing algorithm is not geophysically meaningful: e.g., it predicts velocities outside of the range of reasonable velocities at this depth. To avoid such situations, it is desirable to incorporate the expert knowledge into the algorithm for solving the inverse problem.

We will describe our new result of how to do it.

5.1.2 New Result: Incorporate the Expert Knowledge into the Algorithm for Solving the Inverse Problem

Case of interval prior knowledge. For each cell j, a geophysicist often provides us with his or her estimate of possible values of the corresponding slowness s_j. Sometimes,

this estimates comes in the form of an interval $[\underline{s}_j, \overline{s}_j]$ that is guaranteed to contain the (unknown) actual value of slowness.

It is desirable to modify Hole's algorithm in such a way that on all iterations, slownesses s_j stay within the corresponding intervals. Such a modification is described in [7, 8].

Namely, in the original Hole's algorithm, once we know the current approximations $s_j^{(k)}$ to slownesses, then, along each path i, among all corrections Δs_{ij} that provide the desired compensation, i.e.., for which

$$\sum_{j=1}^{c} \ell_{ij} \cdot \Delta s_{ij} = \Delta t_i, \tag{5.1.1}$$

we find the assignment that minimizes the objective function $\sum\limits_{j,j'}(\Delta s_{ij} - \Delta s_{ij'})^2$, i.e., equivalently, that minimizes the variance of the values Δs_{ij} along this path:

$$V \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{j=1}^{c} \Delta s_{ij}^2 - \left(\frac{1}{n} \cdot \sum_{j=1}^{c} \Delta s_{ij} \right)^2. \tag{5.1.2}$$

In the presence of the interval prior information, on each iteration of Hole's algorithm, we must still minimize the objective function (5.1.2), but this time, we minimize it under two constraint: the same constraint (5.1.1) and the new constraints

$$\underline{s}_j \le s_j^{(k)} + \Delta s_{ij} \le \overline{s}_j. \tag{5.1.3}$$

We have found the following efficient $O(c \cdot \log(c))$ time algorithm for solving the corresponding constraint optimization problem. We start with the initial slowness values $s_j^{(0)}$ which are within the given intervals $[\underline{s}_j, \overline{s}_j]$.

On each iteration of the new procedure, we start with the slowness values $s_j^{(k-1)}$ which are within given intervals $[\underline{s}_j, \overline{s}_j]$. Based on these slownesses, we find the paths from the sources to the sensors, compute the predicted travel-times t_i along each path, and the discrepancies $\Delta t_i = \widetilde{t}_i - t_i$.

We then compute, for each cell j, the values $\underline{\Delta}_j = \underline{s}_j - s_j^{(k-1)}$ and $\overline{\Delta}_j = \overline{s}_j - s_j^{(k-1)}$. We will consider the case when $\Delta t_i > 0$; the case when $\Delta t_i < 0$ is treated similarly. In this case, we first sort all c values $\overline{\Delta}_j$ along the i-th path into a non-decreasing sequence

$$\overline{\Delta}_{(1)} \le \overline{\Delta}_{(2)} \le \ldots \le \overline{\Delta}_{(c)}.$$

Then, for every p from 0 to c, we compute the values A_p and $\mathcal{L}_p$ as follows:

$$A_0 = 0, \quad \mathcal{L}_0 = L_i, \quad A_p = A_{p-1} + \ell_{i(p)} \cdot \overline{\Delta}_{(p)}, \quad \mathcal{L}_p = \mathcal{L}_{p-1} - \ell_{i(p)}.$$

After that, for each p, we compute $S_p = A_p + \mathcal{L}_p \cdot \Delta_{(p+1)}$, and we find p for which $S_{p-1} \le \Delta t_i < S_p$. Once this p is found, we take $\Delta s_{i(j)} = \overline{\Delta}_j$ for $j \le p$, and for $j > p$, we take $\Delta s_{(j)} = \dfrac{\Delta t_i - A_p}{\mathcal{L}_p}$.

When $\Delta t_i < 0$, we similarly sort the values $\underline{\Delta}_j$ into a decreasing sequence, and find p so that the first p corrections are "maxed out" to $\underline{\Delta}_j$, and the rest $c - p$ corrections are determined from the condition $\Delta s_{i(j)} = \dfrac{\Delta t_i - A_p}{\mathcal{L}_p}$.

Once we have computed these corrections for all the paths, then for each cell j, we take the average (or weighted average) of all the corrections coming from all the paths which pass through this cell.

Case of fuzzy prior knowledge. In general, experts are often not 100% sure about the corresponding intervals. They can usually produce a wider interval $[\underline{s}_j, \overline{s}_j]$ of which they are practically 100% certain, but in addition to that, they can also produce narrower intervals about which their degree of certainty is smaller. As a result, instead of a single interval, we have a nested family of intervals corresponding to different levels of uncertainty – i.e., in effect, a fuzzy interval (of which different intervals are α-cuts).

So, instead of simply saying that a given solution to the seismic inverse problem is satisfying or not, we provide a *degree* to which the given solution is satisfying – as the largest α for which the velocity at every point is within the corresponding α-cut intervals.

To solve the seismic inverse problem under such fuzzy uncertainty, we apply the interval algorithm for α-cuts corresponding to $\alpha = 0$, $\alpha = 0.1$, $\alpha = 0.2$, etc., until we reach such a value of α that the process no longer converges. Then, the solution corresponding to the previous value α – i.e., to the largest value α for which the process converged – is returned as the desired solution to the seismic inverse problem.

Case of probabilistic prior knowledge. Often, prior information comes from processing previous observations of the region of interest. In this case, before our experiments, for each cell j, we know a prior (approximate) slowness value $\widetilde{s}_j$, and we know the accuracy (standard deviation) σ_j of this approximate value $\widetilde{s}_j$. It is known that this prior information can lead to much more accurate velocity models; see, e.g., [128]. How can we modify Hole's algorithm so that it takes this prior information into account?

Due to the prior knowledge, for each cell j, the ratio $\dfrac{(s_j^{(k)} + \Delta s_{ij}) - \widetilde{s}_j}{\sigma_j}$ is normally distributed with 0 mean and variance 1. Since each path i consists of a reasonable number of cells, we can thus conclude that the sample variance of this ratio should be close to σ_j, i.e., that

$$\frac{1}{n} \cdot \sum_{j=1}^{c} \frac{((s_j^{(k)} + \Delta s_{ij}) - \widetilde{s}_j)^2}{\sigma_j^2} = 1. \tag{5.1.4}$$

So, to find the corrections Δs_{ij}, we must minimize the objective function (5.1.2) under the constraints (5.1.1) and (5.1.4).

By applying the Lagrange multiplier method to this problem, we can reduce this problem to the unconstrained minimization problem

$$\frac{1}{n} \cdot \sum_{j=1}^{c} \Delta s_{ij}^2 - \left(\frac{1}{n} \cdot \sum_{j=1}^{c} \Delta s_{ij}\right)^2 + \lambda \cdot \left(\sum_{j=1}^{c} \ell_{ij} \cdot \Delta s_{ij} - \Delta t_i\right) +$$

$$\mu \cdot \frac{1}{n} \cdot \sum_{j=1}^{c} \frac{(s_j^{(k)} + \Delta s_{ij} - \widetilde{s}_j)^2}{\sigma_j^2} \to \min. \tag{5.1.5}$$

Differentiating this equation by Δs_{ij} and equating the derivative to 0, we conclude that

$$\frac{2}{n} \cdot \Delta s_{ij} - \frac{2}{n} \cdot \overline{\Delta s} + \lambda \cdot \ell_{ij} + \frac{2\mu}{n \cdot \sigma_j^2} \cdot (s_j^{(k)} + \Delta s_{ij} - \widetilde{s}_j) = 0,$$

where

$$\overline{\Delta s} \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{j=1}^{c} \Delta s_{ij}. \tag{5.1.6}$$

Once we fix λ, μ, and $\overline{\Delta s}$, we get an explicit expression for the values Δs_{ij}. Substituting these expressions into the equations (5.1.1), (5.1.4), and (5.1.6), we get an easy-to-solve

system of 3 non-linear equations with 3 unknowns, which we can solve, e.g., by using Newton's method.

Now, instead of explicit formulas for a transition from $s_j^{(k)}$ to $s_j^{(k+1)}$, we need a separate iteration process – so the computation time is somewhat larger, but we get a more geophysically meaningful velocity map – that takes prior knowledge into account.

Combination of different types of prior knowledge. In many real-life situations, we have both the prior measurement results – which lead to the probabilistic prior knowledge, and expert estimates – which lead to interval and fuzzy prior knowledge. In the presence of probabilistic and interval prior knowledge, we must minimize (5.1.2) under the constraints (5.1.1), (5.1.3), and (5.1.4).

If we replace the equality in (5.1.4) by an inequality ≤ 1, then we get a problem of minimizing a convex function under convex constraints, a problem for which there are known efficient algorithms; see, e.g., [201].

For example, we can use a method of alternating projections, in which we first add a correction that satisfy the first constraint, then the additional correction that satisfies the second constraint, etc. In our case, we first add equal values of Δs_{ij} to satisfy the constraint (5.1.2), then we restrict the values to the nearest points from the interval $[\underline{s}_j, \overline{s}_j]$ – to satisfy the constraint (5.1.3), and after that, find the extra corrections that satisfy the condition (5.1.4), after which we repeat the cycle again until the process converges.

5.2 Computer Engineering

In chip design, one of the main objectives is to decrease its clock cycle. On the design stage, this time is usually estimated by using worst-case (interval) techniques, in which we only use the bounds on the parameters that lead to delays. This analysis does not take into account that the probability of the worst-case values is usually very small; thus, the resulting estimates are over-conservative, leading to unnecessary over-design and under-performance

of circuits. If we knew the *exact* probability distributions of the corresponding parameters, then we could use Monte-Carlo simulations (or the corresponding analytical techniques) to get the desired estimates. In practice, however, we only have *partial* information about the corresponding distributions, and we want to produce estimates that are valid for all distributions which are consistent with this information.

In this dissertation, we will develop a general technique that allows us, in particular, to provide such estimates for the clock time.

5.2.1 Problem of Decreasing Clock Cycle: A Brief Introduction

Decreasing clock cycle: a practical problem. In chip design, one of the main objectives is to decrease the chip's clock cycle. It is therefore important to estimate the clock cycle on the design stage.

The clock cycle of a chip is constrained by the maximum path delay over all the circuit paths $D \stackrel{\text{def}}{=} \max(D_1, \ldots, D_N)$, where D_i denotes the delay along the i-th path. Each path delay D_i is the sum of the delays corresponding to the gates and wires along this path. Each of these delays, in turn, depends on several factors such as the variation caused by the current design practices, environmental design characteristics (e.g., variations in temperature and in supply voltage), etc.

Traditional (interval) approach to estimating the clock cycle. Traditionally, the delay D is estimated by using the worst-case analysis, in which we assume that each of the corresponding factors takes the worst possible value (i.e., the value leading to the largest possible delays). As a result, we get the time delay that corresponds to the case when all the factors are at their worst.

It is necessary to take probabilities into account. The worst-case analysis does not take into account that different factors come from independent random processes. As a result, the probability that all these factors are at their worst is extremely small. For

example, there may be slight variations of delay time from gate to gate, and this can indeed lead to gate delays. The worst-case analysis considers the case when all these random variations lead to the worst case; since these variations are independent, this combination of worst cases is highly unprobable.

As a result, the current estimates of the chip clock time are over-conservative, over up to 30% above the observed clock time. Because of this over-estimation, the clock time is set too high – i.e., the chips are usually over-designed and under-performing; see, e.g., [26, 37, 38, 162, 161, 163, 164]. To improve the performance, it is therefore desirable to take into account the probabilistic character of the factor variations.

Robust statistical methods are needed. If we knew the *exact* probability distributions of the corresponding parameters, then we could use Monte-Carlo simulations (or the corresponding analytical techniques) to get the desired estimates. In practice, however, we only have *partial* information about the corresponding distributions. For a few parameters, we know the exact distribution, but for most parameters, we only know the mean and some characteristic of the deviation from the mean – e.g., the interval that is guaranteed to contain possible values of this parameter.

In principle, we could pick up some distributions which are consistent with this partial information – e.g., truncated normal distributions, compute the maximum delays D corresponding to all these distributions, and then take the largest $D_{\max}$ of these computed maximum delays D as the clock time. This procedure will guarantee that the path delay D does not exceed the clock time if the actual distribution is one of the picked ones. However, it is quite possible that some other possible distributions (different from the ones we picked), the corresponding path delay D is larger than $D_{\max}$. As a result, we may be underestimating the clock time. If we set the clock time too low, we may have operations that did not have time to finish before the next cycle starts – and this is even worse than overestimating.

It is therefore desirable to provide bounds that work for all the distributions which are

consistent with the given information. In statistics, estimates which are guaranteed for all distributions from some non-parametric class are called *robust* (see, e.g., [85]). In these terms, our objective is to provide robust statistical estimates for the clock time.

5.2.2 New Result: General Techniques to Provide Robust Estimates for the Clock Time.

In this dissertation, we will develop general techniques that allow us, in particular, to provide robust estimates for the clock time.

In deriving these estimates, we will use the extensions of interval methods to cases with partial information about probabilities described, e.g., in [64, 111, 108, 109]; see also [21, 22, 23, 24, 124].

Towards a mathematical formulation of the problem. 1°. How the desired delay d depends on the parameters? The variations in the each gate delay d are caused by the difference between the actual and the nominal values of the corresponding parameters. It is therefore desirable to describe the resulting delay d as a function of these differences $x_1, \ldots, x_n$. Since these differences are usually small, we can safely ignore quadratic (and higher order) terms in the Taylor expansion of the dependence of d on x_j and assume that the dependence of each delay d on these differences can be described by a linear function.

As a result, each path delay D_i – which, as we have mentioned, is the sum of delays at different gates and wires – can also be described as a linear function of these differences, i.e., as $D_i = a_i + \sum_{j=1}^{n} a_{ij} \cdot x_j$ for some coefficients a_i and a_{ij}.

Thus, the desired maximum delay $D = \max_i D_i$ has the form

$$D = \max_i \left(a_i + \sum_{j=1}^{n} a_{ij} \cdot x_j \right). \tag{5.2.1}$$

2°. How we can describe such functions in general terms? In this dissertation, we will use two properties of the time delay. First, we will use the fact that the time delay is always non-negative; second, we will use the fact that the dependence (5.2.1) is convex.

Let us recall that a function $f : R^m \to R$ is called *convex* if

$$f(\alpha \cdot x + (1-\alpha) \cdot y) \le \alpha \cdot f(x) + (1-\alpha) \cdot f(y)$$

for every $x, y \in R^m$ and for every $\alpha \in (0,1)$. It is known that the maximum of several linear functions is convex, so the function (5.2.1) is convex. Vice versa, every convex function can be approximated, with an arbitrary accuracy, by maxima of linear functions – i.e., by expressions of type (5.2.1).

So, in general terms, we can say that we are interested in the robust statistical properties of the value $y = F(x_1, \ldots, x_n)$, where F is a non-negative convex function of the variables x_j.

3°.Our objective: We want to find the smallest possible value y_0 such that for all possible distributions consistent with the known information, we have $y \le y_0$ with the probability $\ge 1 - \varepsilon$ (where $\varepsilon > 0$ is a given small probability).

4°. What information can we use for these estimations? We can safely assume that different factors x_j are statistically independent. About some of the variables x_j, we know their exact statistical characteristics; about some other variables x_j, we only know their interval ranges $[\underline{x}_j, \overline{x}_j]$ and their means E_j.

5°. There is additional property: the dependency is non-degenerate. We only have partial information about the probability distribution of the variables x_j. For each possible probability distribution p, we can find the largest value y_p for which, for this distribution, $y \le y_p$ with probability $\ge 1 - \varepsilon$. The desired value y_0 is the largest of the values y_p corresponding to different probability distributions p: $y_0 = \sup\limits_{p \in \mathcal{P}} y_p$, where $\mathcal{P}$ denotes the class of probability distributions p which are consistent with the known information.

If we learn some additional information about the distribution of x_j – e.g., if we learn that x_j actually belongs to a proper subinterval of the original interval $[\underline{x}_j, \overline{x}_j]$ – we thus

decrease the class $\mathcal{P}$ of distributions p which are consistent with this information, to a new class $\mathcal{P}' \subset \mathcal{P}$. Since the class has decreased, the new value $y_0' = \sup\limits_{p \in \mathcal{P}'} y_p$ is the maximum over a smaller set and thus, cannot be larger than the original value y_0: $y_0' \le y_0$.

From the purely mathematical viewpoint, it is, in principle, possible that the desired value y does not actually depend on some of the variables x_j. In this case, if we narrow down the interval of possible values of the corresponding variable x_j, this will not change the resulting value y_0.

For the chip design problem, it is reasonable to assume that such variables have already been weeded out, and that the resulting function $F(x_1, \ldots, x_n)$ is *non-degenerate* in the sense that every time we narrow down one of the intervals $[\underline{x}_j, \overline{x}_j]$, the resulting value y_0 actually decreases: $y_0' < y_0$.

As a result, we arrive at the following problem.

Formulation of the problem and the main result.

GIVEN:
- natural numbers n, and $k \le n$;
- a real number $\varepsilon > 0$;
- a function $y = F(x_1, \ldots, x_n)$ (algorithmically defined) such that for every combination of values $x_{k+1}, \ldots, x_n$, the dependence of y on $x_1, \ldots, x_k$ is convex;
- $n - k$ probability distributions $x_{k+1}, \ldots, x_n$ – e.g., given in the form of cumulative distribution function (cdf) $F_j(x)$, $k + 1 \le j \le n$;
- k intervals $\mathbf{x}_1, \ldots, \mathbf{x}_k$, and
- k values $E_1, \ldots, E_k$,

such that for every $x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, x_k \in [\underline{x}_k, \overline{x}_k]$, we have $F(x_1, \ldots, x_n) \ge 0$ with probability 1.

TAKE: all possible joint probability distributions on R^n for which:
- all n random variables are independent;

- for each j from 1 to k, $x_j \in \mathbf{x}_j$ with probability 1 and the mean value of x_j is equal to E_j;
- for $j > k$, the variable x_j has a given distribution $F_j(x)$.

FIND: find the smallest possible value y_0 such that for all possible distributions consistent with the known information, we have $y \stackrel{\text{def}}{=} F(x_1, \ldots, x_n) \le y_0$ with probability $\ge 1 - \varepsilon$.

PROVIDED: that the problem is *non-degenerate* in the sense that if we narrow down one of the intervals $\mathbf{x}_j$, the value y_0 decreases.

Towards an algorithm for computing y_0. The following result explains how we can compute this value y_0.

Proposition 1 *The desired value y_0 is attained when for each j from 1 to k, we use a 2-point distribution for x_j, in which:*

- *$x_j = \underline{x}_j$ with probability $\underline{p}_j \stackrel{\text{def}}{=} \dfrac{\overline{x}_j - E_j}{\overline{x}_j - \underline{x}_j}$.*
- *$x_j = \overline{x}_j$ with probability $\overline{p}_j \stackrel{\text{def}}{=} \dfrac{E_j - \underline{x}_j}{\overline{x}_j - \underline{x}_j}$.*

The proof of Proposition 1 is as follows.

By definition, y_0 is the largest value of y_p over all possible distributions $p \in \mathcal{P}$. This means that for the given y_0, for all possible distributions $p \in \mathcal{P}$, we have $\text{Prob}(D \le y_0) \ge 1 - \varepsilon$. Let $p \in \mathcal{P}$ be the "worst-case" distribution, i.e., the distribution for which the probability $\text{Prob}(D \le y_0)$ is the smallest. Let us show that this "worst case" occurs when all k variables $x_1, \ldots, x_k$ have the 2-point distributions described in Proposition 1.

Let us fix the value $j \le k$ and show that in the "worst case", x_j indeed has the desired 2-point distribution. Without losing generality, we can take $j = 1$. Let us fix the distributions for $x_2, \ldots, x_k$ as in the worst case. Then, the fact that the probability $\text{Prob}(D \le y_0)$ is the smallest means that if we replace the worst-case distribution for x_1

with some other distribution, we can only increase this probability. In other words, when we correspondingly fix the distributions for $x_2, \ldots, x_k$, the probability $\mathrm{Prob}(D \le y_0)$ attains the smallest possible value at the desired distribution for x_1.

In reality, the distribution for x_1 is located on an interval $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$, i.e., on a set with infinitely many points. However, with an arbitrary large value N (and thus, for an arbitrarily small discretization error $\delta = (\overline{x}_1 - \underline{x}_1)/N$), we can assume that all the distributions are located on a finite grid of values

$$v_0 \stackrel{\mathrm{def}}{=} \underline{x}_1, \quad v_1 \stackrel{\mathrm{def}}{=} \underline{x}_1 + \delta, \quad v_2 \stackrel{\mathrm{def}}{=} \underline{x}_1 + 2\delta, \ldots, v_N = \overline{x}_1.$$

The smaller δ, the better this approximation. Thus, without losing generality, we can assume that the distribution of x_1 is located on finitely many points v_i.

In this approximation, the probability distribution for x_1 can be described by the probabilities $q_i \stackrel{\mathrm{def}}{=} p_1(v_i)$ of different values v_i.

The minimized probability $\mathrm{Prob}(D \le y_0)$ can be described as the sum of the probabilities of different combinations $(x_1, \ldots, x_n)$ over all the combinations for which $D(x_1, \ldots, x_n) \le y_0$. We assumed that all the variables x_j are independent. Thus, the probability of each combination $(x_1, \ldots, x_n)$ is equal to the product of the corresponding probabilities $p_1(x_1) \cdot p_2(x_2) \cdot \ldots$ Since the probability distributions for $x_2, \ldots$ are fixed, the minimized probability is thus a linear combination of probabilities $p_1(v_i)$, i.e., of the probabilities q_i. In other words, the minimized probability has the form $\sum\limits_{i=0}^{N} c_i \cdot q_i$ for some coefficients c_i.

By describing the probability distribution on x_1 via the probabilities $q_i = p_1(v_i)$ of different values $v_i \in [\underline{x}_1, \overline{x}_1]$, we automatically restrict ourselves to distributions which are located on this interval. The only restrictions that we have on the probability distribution of x_1 is that it is a probability distribution, i.e., that $q_i \ge 0$ for all i and $\sum\limits_{i=0}^{N} q_i = 1$, and that the mean value of this distribution is equal to E_1, i.e., that $\sum\limits_{i=0}^{N} q_i \cdot v_i = E_1$. Thus, the worst-case distribution for x_1 is a solution to the following linear programming problem:

Minimize

$$\sum_{i=0}^{N} c_i \cdot q_i$$

under the constraints

$$\sum_{i=0}^{N} q_i = 1,$$

$$\sum_{i=0}^{N} q_i \cdot v_i = E_1,$$

$$q_i \geq 0, \quad i = 0, 1, 2, \ldots, N.$$

It is known that the solution to the linear programming problem is always attained at a vertex of the corresponding constraint set. In other words, in the solution to the linear programming problem with $N + 1$ unknowns $q_0, q_1, \ldots, q_N$, at least $N + 1$ constraints are equalities. Since we already have 2 equality constraints, this means that out of the remaining constraints $q_i \geq 0$, at least $N - 1$ are equalities. In other words, this means that in the optimal distribution, all but two values of $q_i = p_1(v_i)$ are equal to 0.

Thus, the "worst-case" distribution for x_1 is located on 2 points v and v' within the interval $[\underline{x}_1, \overline{x}_1]$. Let us prove, by reduction to a contradiction, that these two points cannot be different from the endpoints of this interval. Indeed, let us assume that they are different. Without losing generality, we can assume that $v \leq v'$. Then, this "worst-case" distribution is actually located on the proper subinterval $[v, v'] \subset [\underline{x}_1, \overline{x}_1]$ of the original interval $\mathbf{x}_1$. Since the maximum y_0 of y_p is attained on this distribution, replacing the original interval $\mathbf{x}_1$ with its proper subinterval $[v, v']$ would not change the value y_0 – while our assumption of non-degeneracy states that such a replacement would always lead to a smaller value y_0. This contradiction shows that the values v and v' – on which the worst-case distribution is located – have to be endpoints of the interval $[\underline{x}_1, \overline{x}_1]$.

In other words, we conclude that the worst-case distribution is located at 2 points: $\underline{x}_1$ and $\overline{x}_1$. Such a distribution is uniquely determined by the probabilities $\underline{p}_1$ and $\overline{p}_1$ of these two points. Since the sum of these probabilities is equal to 1, it is sufficient to describe one of these probabilities, e.g., $\overline{p}_1$; then, $\underline{p}_1 = 1 - \overline{p}_1$. The condition that the mean of x_1 is E_1, i.e., that

$$\underline{p}_1 \cdot \underline{x}_1 + \overline{p}_1 \cdot \overline{x}_1 = (1 - \overline{p}_1) \cdot \underline{x}_1 + \overline{p}_1 \cdot \overline{x}_1 = E_1,$$

uniquely determines $\overline{p}_1$ (and hence $\underline{p}_1$) – exactly by the expression from Proposition 1. The statement is proven.

Resulting algorithm for computing y_0. Because of Proposition 1, we can compute the desired value y_0 by using the following Monte-Carlo simulation:

- We set each value x_j, $1 \le j \le k$, to be equal to $\overline{x}_j$ with probability $\overline{p}_j$ and to the value $\underline{x}_j$ with the probability $\underline{p}_j$.
- We simulate the values x_j, $k < j \le n$, as random variables distributed according to the distributions $F_j(x)$.
- For each simulation s, $1 \le s \le N_i$, we get the simulated values $x_j^{(s)}$, and then, a value $y^{(s)} = F(x_1^{(s)}, \ldots, x_n^{(s)})$. We then sort the resulting N_i values $y^{(s)}$ into an increasing sequence

  $$y_{(1)} \le y_{(2)} \le \ldots \le y_{(N_i)},$$

  and take, as y_0, the $N_i \cdot (1-\varepsilon)$-th term $y_{(N_i \cdot (1-\varepsilon))}$ in this sorted sequence.

Comment about Monte-Carlo techniques. Let us remark that some readers may feel uncomfortable with the use of Monte-Carlo techniques. This discomfort comes from the fact that in the *traditional* statistical approach, when we know the exact probability distributions of all the variables, Monte-Carlo methods – that simply simulate the corresponding distributions – are inferior to analytical methods. This inferiority is due to two reasons:

- First, by design, Monte-Carlo methods are approximate, while analytical methods are usually exact.
- Second, the accuracy provided by a Monte-Carlo method is, in general, proportional to $\sim 1/\sqrt{N_i}$, where N_i is the total number of simulations. Thus, to achieve reasonable quality, we often need to make a lot of simulations – as a result, the computation time of a Monte-Carlo method becomes much longer than the time of an analytical method.

In *robust* statistic, there is often an additional reason to be uncomfortable about using Monte-Carlo methods:

- Practitioners use these methods by selecting a finite set of distributions from the infinite class of all possible distributions, and running simulations for the selected distributions.
- Since we do not test all the distributions, this practical heuristic approach sometimes misses the distributions on which the minimum or maximum of the corresponding distribution is actually attained.

In our case, we also select a finite collection of distributions from the infinite set. However, in contrast to the heuristic (un-justified) selection – which is prone to the above criticism, our selection is *justified*. Proposition 1 *guarantees* that the values corresponding to the selected distributions indeed provide the desired value y_0 – the largest over all possible distributions $p \in \mathcal{P}$.

In such situations, where a justified selection of Monte-Carlo methods is used to solve a problem of robust statistics, such Monte-Carlo methods often lead to *faster* computations than known analytical techniques. The speed-up caused by using such Monte-Carlo techniques is one of the main reasons why they were invented in the first place – to provide fast estimates of the values of multi-dimensional integrals. Many examples of efficiency of these techniques are given, e.g., in [177]; in particular, examples related to estimating how the uncertainty of inputs leads to uncertainty of the results of data processing are given in [200].

Comment about Non-linear Terms. In the formula (5.3.1), we ignored quadratic and higher order terms in the dependence of each path time D_i on the parameters x_j. It is known that the maximum $D = \max\limits_i D_i$ of convex functions D_i is always convex. So, according to Proposition 1, the above algorithm will work if we take quadratic terms into consideration – provided that each dependence $D_i(x_1, \ldots, x_k, \ldots)$ is still convex.

5.3 Radar Data Processing

A radar observes the result of a space explosion. Due to radar's low horizontal resolution, we get a 1-D signal $x(t)$ representing different 2-D slices. Based on these slices, we must distinguish between the body at the core of the explosion and the slowly out-moving fragments. We propose new algorithms for processing this 1-D data. Since these algorithms are time-consuming, we also exploit the possibility of parallelizing these algorithms.

5.3.1 Formulation of the Problem

Most astronomical processes are slow; however, sometimes, space explosions happen: starts become supernovae, planetoids are torn apart by tidal and gravitational forces, etc. Even the Universe itself is currently viewed as a result of such an explosion – the Big Bang.

From the astrophysical viewpoint, these explosions are very important, because, e.g., supernovae explosions is how heavy metals spread around in the Universe.

The explosion processes are very rare and very fast, so unless they are very powerful and spectacular – like an explosion of a nearby supernovae that happened in 1054 – they are very difficult to observe. As a result, space explosion processes often go unnoticed.

What we do observe in most cases is the *result* of the space explosion, i.e., the explosion core – the remainder of the original celestial body – surrounded by the explosion fragments. The most well known example of such a result is the Crab Nebula formed after the 1054 supernovae explosion.

In order to better understand the corresponding physical process, it is extremely important to identify the explosion core.

In space, there is not much friction, so, due to inertia, most of the fragments travel with approximately the same speed as in the beginning of the explosion. Dividing the distance between the two fragments by their relative speed, we can determine – reasonably accurately – when the explosion occurred (this is how we know that the supernovae in the Crab Nebulae exploded in the year 1054). At that explosion time, all the fragments and

the core were located at the same point, so it is difficult to distinguish between the core and the fragments.

In general, we have a 2-D (and sometimes even 3-D) image of the result of the explosion. In such situations, detecting the explosion core is an image processing problem.

However, there is one important case when we only have 1-D data. In this case, we cannot use image processing techniques, we have to use techniques for processing 1-D data – i.e., DSP techniques.

This is the case of nearby space explosions, when the radar is the main source of information. A radar sends a pulse signal toward an object; this signal reflects from the object back to the station. We can measure, very accurately, the overall time that the signal traveled, which gives us the distance to the object. We can also measure the velocity, or, to be more precise, the rate with which the distance changes. It is, however, very difficult to separate the signals from different fragments located at the same distance.

As a result, what we observe is a 1-D signal $s(t)$, where each value $s(t)$ represents the intensity of the reflection from all the fragments located at distance $c \cdot t$ from the radar – i.e., from the 2-D slice corresponding to this distance. Based on these slices, we must distinguish between the body at the core of the explosion and the (slowly expanding) fragments.

We will describe a new method of identifying a core based on the slice observations.

5.3.2 New Result: A New Method for Solving the Problem

Repeated signal measurements at several different moments of time T_k. At first glance, there may seem to be no difference between the signals reflected by the fragments and the signal reflected by the core. However, in the process of an explosion, fragments usually start rotating fast, at random rotation frequencies, with random phases. As a result, the signals reflected from the fragments oscillate, while the signal from the original core practically does not change.

As a result, the reflected signals change with time. Therefore, it makes sense to measure the signal $s(t)$ not just once, but at several consequent moments of time, i.e., to consider

the signals $s_1(t), \ldots, s_N(t)$ measured at moments $T_1 < \ldots < T_N$, and use the difference between the dynamic character of the fragments and the static character of the core to identify the core.

Relating measurements performed at different moments of time $T_k \neq T_l$: the corresponding t-scales are linearly related. In order to compare signals measured at different moments of time $T_k \neq T_l$, we must identify the layers measured at different moments of time.

Let T_0 be the moment of explosion, and let x_0 be the initial distance between the radar and the core (and the fragments) at that initial moment of time T_0. We assume that our coordinate system has the radar as its origin, and that the x axis is the axis in the direction of the analyzed "cloud". For each fragment i, let $v_x^{(i)}$ be the x-component of the velocity of i-th fragment (velocity relative to the radar). Hence, at moment T_k, the x-coordinate of i-th fragment in our coordinate system – i.e., its distance from the radar – is equal to $x^{(i)}(T_k) = x_0 + v_x^{(i)} \cdot (T_k - T_0)$. Therefore, the radar signal reflected from this fragment corresponds to the time

$$t_k^{(i)} = \frac{x_k^{(i)}}{c} = \frac{x_0}{c} + v_x^{(i)} \cdot \frac{T_k - T_0}{c}. \tag{5.3.1}$$

Similarly, when we repeat the radar measurement at time $T_l \neq T_k$, the radar signal reflected from the i-th fragment corresponds to the time

$$t_l^{(i)} = \frac{x_0}{c} + v_x^{(i)} \cdot \frac{T_l - T_0}{c}. \tag{5.3.2}$$

What is the relation between the corresponding times $t_k^{(i)}$ and $t_l^{(i)}$? From the equation (5.3.1), we conclude that

$$v_x^{(i)} = \frac{c \cdot t_k^{(i)} - x_0}{T_k - T_0}.$$

Substituting this expression into the formula (5.3.2), we conclude that

$$t_l^{(i)} = \frac{x_0}{c} + \frac{c \cdot t_k^{(i)} - x_0}{T_k - T_0} \cdot \frac{T_l - T_0}{c} = a_{kl} \cdot t_k^{(i)} + b_{kl}, \tag{5.3.3}$$

where

$$a_{kl} = \frac{T_l - T_0}{T_k - T_0} > 0$$

and

$$b_{kl} = \frac{x_0}{c} - \frac{x_0}{T_k - T_0} \cdot \frac{T_l - T_0}{c}$$

do not depend on i.

In other words, the t-scales of the signals $s_k(t)$ and $s_l(t)$ are related by a linear dependence $t_k \to t_l = a_{kl} \cdot t_k + b_{kl}$.

How can we experimentally find the coefficients of this linear relation? At each moment of time T_k, we get the observed signal $s_k(t)$. Let $\underline{t}_k$ be the smallest time at which we get some reflection from the fragments cloud, and let $\overline{t}_k$ be the largest time at which we observe the radar reflection from this cloud. This means that there is a fragment i for which $t_k^{(i)} = \underline{t}_k$, there is a fragment j for which $t_k^{(j)} = \overline{t}_k$, and for every other fragment f, the corresponding moment of time is in between $\underline{t}_k$ and $\overline{t}_k$: $t_k^{(f)} \in [\underline{t}_k, \overline{t}_k]$.

As we have mentioned, for every other observation T_l, the relation between the corresponding times $t_k^{(i)}$ and $t_l^{(i)}$ is linear, with a positive coefficient a_{kl}. Since $a_{kl} > 0$, the corresponding linear functions $t \to a_{kl} \cdot t + b_{kl}$ is monotonically increasing. Thus, the value t_l is the smallest for the same fragment i for which t_k was the smallest. Hence, $\underline{t}_l = t_l^{(i)} = a_{kl} \cdot t_k^{(i)} + b_{kl}$, i.e.,

$$\underline{t}_l = a_{kl} \cdot \underline{t}_k + b_{kl}. \tag{5.3.4}$$

Similarly,

$$\overline{t}_l = a_{kl} \cdot \overline{t}_k + b_{kl}. \tag{5.3.5}$$

The values $\underline{t}_k$, $\overline{t}_k$, $\underline{t}_l$, and $\overline{t}_l$ are directly observable. Thus, by solving the system of two linear equations (5.3.4) and (5.3.5) with 2 unknowns, we get explicit expressions for a_{kl} and b_{kl} in terms of these observable values:

$$a_{kl} = \frac{\overline{t}_l - \underline{t}_l}{\overline{t}_k - \underline{t}_k}; \quad b_{kl} = \frac{\overline{t}_k \cdot \underline{t}_l - \underline{t}_k \cdot \overline{t}_l}{\overline{t}_k - \underline{t}_k}.$$

How Can We Transform Signals $s_k(t)$ and $s_l(t)$ to the Same Scale? Our main idea is that after we measure the fragments cloud at two different moments of time T_k and T_l, we should compare the values $s_k(t)$ and $s_l(t)$ corresponding to the same fragments.

We know that for each moment of time t, the value $s_k(t)$ describes the same fragment(s) as the value $s_l(t')$, where $t' = a_{kl} \cdot t + b_{kl}$. We also know how to experimentally determine the coefficients a_{kl} and b_{kl}. So, to make the desired comparison easier, it is reasonable to "re-scale" the signals to the same t-scale, so that the compared values correspond to exactly the same value t. In other words, we would like to generate a re-scaled signal

$$\widetilde{s}_l(t) \stackrel{\text{def}}{=} s_l(a_{kl} \cdot t + b_{kl}). \tag{5.3.6}$$

If the measurements were absolutely accurate, i.e., if we had the values $s_k(t)$ corresponding to each individual time t, then such a re-scaling would be easy: we could simply explicitly use the formula (5.3.6).

In real life, however, each value $s_l(t)$ corresponds not just to a single time t, but to the entire "bin" of values, from some value $\underline{t}$ to the value $\underline{t} + \Delta t$, where Δt is the accuracy with which the radar can measure the time t (in other words, $\Delta t = \Delta x/c$, where Δx is the accuracy with which the radar can measure the distance). In other words, what we actually observe is a sequence of values $\ldots$, $s((i-1)\cdot\Delta t)$, $s(i\cdot\Delta t)$, $s((i+1)\cdot\Delta)$, $\ldots$ Crudely speaking, each observed value $s(i \cdot \Delta t)$ represent the overall intensity of all the fragments for which the actual reflection time $t = x/c$ is in the interval

$$I_i \stackrel{\text{def}}{=} [(i - 0.5) \cdot \Delta t, (i + 0.5) \cdot \Delta t]. \tag{5.3.7}$$

Because of this discreteness, we cannot directly use the formula (5.3.6) to match the signals: Indeed, from the moment T_k to the moment T_l, the cloud slightly expands. At the moment T_k, the value $s_k(i \cdot \Delta t)$ is the overall intensity of all the fragments for which t_k belongs to the interval (5.3.6) of width Δt. At moment T_l, the times $t_l = a_{kl} \cdot t_k + b_{kl}$ corresponding to these same fragments occupy a wider interval – of width $a_{kl} \cdot \Delta t > \Delta t$. Thus, these fragments are no longer in the same bin, they may be in different bins.

How can we match the values? A natural idea is to use linear extrapolation. In other words, to estimate $\widetilde{s}(t)$ for $t = i \cdot \Delta t$, we apply the linear transformation $a_{kl} \cdot t + b_{kl}$ to the interval I_i. The resulting interval $\widetilde{I}_i$ consists of several parts from different intervals I_j. As $\widetilde{s}_l(t)$, we take a linear combination of the corresponding values $s_l(j \cdot \Delta t)$, with weights proportional to the relative length $|\widetilde{I}_i \cap I_j|/\Delta t$ of the intersection $\widetilde{I}_i \cap I_j$:

$$\widetilde{s}_l(i \cdot \Delta t) \stackrel{\text{def}}{=} \sum_j \frac{|\widetilde{I}_i \cap I_j|}{\Delta t} \cdot s_l(j \cdot \Delta t).$$

For example, if $\widetilde{I}_i$ consists of the entire interval I_j, 0.1 of I_{j-1}, and 0.05 of I_{i-1}, then $\widetilde{s}_l(i \cdot \Delta t)$ is equal to:

$$0.1 \cdot s_l((i-1) \cdot \Delta t) + s_l(i \cdot \Delta t) + 0.05 \cdot s_l((i+1) \cdot \Delta t).$$

In the following text, we will assume that the signals $s_i(t)$ have already been thus rescaled.

Algorithm: main idea. Each layer ("bin") contains several fragments. These fragments oscillate with random (uncorrelated) frequencies and phases; the overall signal $x(t)$ is the sum of the reflections from all these fragments. Due to the central limit theorem, the resulting overall signal $x(t)$ is approximately normally distributed with some mean $E(t)$ and variance $V(t)$.

If a layer only contains fragments, then, due to the independence assumption, $E(t) \approx n(t) \cdot E$ and $V(t) \approx n(t) \cdot V$, where $n(t)$ is the (unknown) number of fragment in layer t, and E and V are the mean and variance corresponding to each fragment. Therefore, for each such layer, $E(t) \approx (E/V) \cdot V(t)$.

For a layer that also contains the core, we have $E(t) \approx E_c + N(t) \cdot E$ and $V(t) \approx N(t) \cdot V$, where E_c is the intensity of the core (since the core is supposed to be not rotating fast, its signal does not change with time, so the corresponding variance is negligible). Thus, for this layer, $E(t) \approx E_c + (E/V) \cdot V(t)$. So, for the core, $E(t)/V(t) \gg E/V$.

Therefore, crudely speaking, our best guess for the core location is the point t for which the ratio $E(t)/V(t)$ is the largest.

This is, of course, a very naive description of the idea. Let us see how this idea can be described in more adequate DSP terms.

Towards a statistically valid algorithm.

1°. What are motivations for the main distribution formula? The intensity $I_i(t)$ of each fragment i depends on time. Let $a_i = \lim\limits_{T\to\infty} T^{-1} \cdot \int\limits_0^T I_i(t)\,dt$ denote the average intensity over time, and let

$$b_i = \lim_{T\to\infty} T^{-1} \cdot \int_0^T (I_i(t) - a_i)^2\,dt.$$

In the ensemble of fragments, let a_0 be the mean of a_i, let A_0 be the variance of a_i, let b_0 be the mean of b_i, and let B_0 be the mean of a_i. Then, according to the main idea, we can assume that $E(t)$ is normally distributed with the mean $n(t)\cdot a_0$ and the variance $n(t)\cdot A_0$, and $V(t)$ is normally distributed with the mean $n(t)\cdot b_0$ and the variance $n(t)\cdot B_0$.

We assumed the layers to be independent. As a result, we arrive at the following formula for the resulting probability distribution:

$$\rho = \prod_{t=1}^{N} \frac{1}{\sqrt{2\pi\cdot n(t)\cdot A_0}} \cdot \exp\left(-\frac{(E(t)-n(t)\cdot a_0)^2}{2n(t)\cdot A_0}\right) \times$$

$$\prod_{t=1}^{N} \frac{1}{\sqrt{2\pi\cdot n(t)\cdot B_0}} \cdot \exp\left(-\frac{(V(t)-n(t)\cdot b_0)^2}{2n(t)\cdot B_0}\right),$$

with the proviso that for the layer $t = t_0$ containing the core, we have $E(t) - E_c - n(t)\cdot a_0$ instead of $E(t) - n(t)\cdot a_0$.

Based on the experimental data $E(t)$ and $V(t)$, we must find estimates for the parameters a_0, b_0, A_0, B_0, $n(t)$, t_0, and E_c – and what we are really interested in is t_0. In accordance with the Maximum Likelihood Method (MLM), we must find the values of these parameters for which $\rho \to \max$. As usual in statistics, it is convenient to replace the problem of maximizing ρ with a mathematically equivalent problem of minimizing a simpler function $\psi \stackrel{\text{def}}{=} -\ln(\rho)$, i.e., in our case,

$$\psi = \sum_{t=1}^{N} \frac{(E(t)-n(t)\cdot a_0)^2}{2n(t)\cdot A_0} + \sum_{t=1}^{N} \frac{(V(t)-n(t)\cdot b_0)^2}{2n(t)\cdot B_0} +$$

$$\sum_{t=1}^{N} \ln(n(t)) + \frac{N}{2} \cdot \log(A_0) + \frac{N}{2} \cdot \log(B_0). \tag{5.3.8}$$

2°. Let us start with the simplest case when we know the values of the parameters a_0, b_0, A_0, and B_0 that describe the distribution of fragments. In this case, differentiating by $n(t)$ and equating the derivative to 0, we conclude that

$$-\frac{1}{2n(t)^2}\left(\frac{E(t)^2}{A_0} + \frac{V(t)^2}{B_0}\right) + \frac{1}{2}\left(\frac{a_0^2}{A_0} + \frac{b_0^2}{B_0}\right) + \frac{1}{n(t)} = 0.$$

The first two terms are approximately independent on the number of fragments $n(t)$, the third term $1/n(t)$ is much smaller (since we have many fragments). So, we can safely ignore the their term and conclude that $n(t) = \|v_t\|/\|v_0\|$, where we denoted

$$v_t \stackrel{\text{def}}{=} \left(\frac{E(t)}{\sqrt{A_0}}, \frac{V(t)}{\sqrt{B_0}}\right); \quad v_0 \stackrel{\text{def}}{=} \left(\frac{a_0}{\sqrt{A_0}}, \frac{b_0}{\sqrt{B_0}}\right),$$

and $\|(v_a, v_b)\| = \sqrt{v_a^2 + v_b^2}$ denotes the length of the vector $v = (v_a, v_b)$. Substituting this expression for $n(t)$ into the corresponding part of (5.3.8), i.e., into

$$\psi(t) \stackrel{\text{def}}{=} \frac{(E(t) - n(t) \cdot a_0)^2}{2n(t) \cdot A_0} + \frac{(V(t) - n(t) \cdot b_0)^2}{2n(t) \cdot B_0} +$$

$$\ln(n(t)) = \frac{1}{2n(t)} \cdot \left(\frac{E(t)^2}{A_0} + \frac{V(t)^2}{B_0}\right) -$$

$$\left(\frac{E(t) \cdot a_0}{A_0} + \frac{V(t) \cdot a_0}{A_0}\right) + \frac{n(t)}{2} \cdot \left(\frac{a_0^2}{A_0} + \frac{b_0^2}{B_0}\right) + \ln(n(t)),$$

we conclude that $\psi(t) \approx \psi_0(t)$, where

$$\psi_0(t) \stackrel{\text{def}}{=} \|v_t\| \cdot \|v_0\| - v_t \cdot v_0, \tag{5.3.9}$$

and $v_t \cdot v_0$ denotes the dot (scalar) product. ($\approx$ because we use the approximate value for $n(t)$.)

For $t = t_0$, due to the presence of an additional variable E_c, we get $\psi(t_0) \approx 0$. Thus,

$$\psi = (N/2) \cdot (\log(A_0) + \log(B_0)) + \sum_{t=1}^{N} \psi_0(t) - \psi_0(t_0).$$

Thus, ψ is the smallest if and only if $\psi(t_0)$ is the largest. Therefore, we arrive at the following algorithm for locating the core:

- First, we re-scale the signals $s_k(t)$ into $\widetilde{s}_k(t)$ so that the same value t corresponds to the same fragments.
- For each t, we compute the sample average $E(t)$ and the sample variance $V(t)$ of the values $\widetilde{s}_k(t)$.
- For each t, we compute v_t and $\psi_0(t)$, and find t_0 for which $\psi_0(t_0) = m \stackrel{\text{def}}{=} \max_t \psi_0(t)$.

How reliable is this estimate? We are interested in the value of a single variable t_0, and we know that for one variable, 95% of the values are within 2σ from the mean, and 99.9% are within 3σ. In terms of $\psi = \ln(\rho)$, the mean corresponds to its minimum, the 2σ deviation means difference $(2\sigma)^2/(2\sigma^2) = 2$ from the minimum, and 3σ deviation means the difference of $(3\sigma)^2/(2\sigma^2) = 4.5$ from the minimum. Thus, with reliability 95%, we conclude that the core is among those t for which $\psi_0(t) \geq m - 2$, and that with reliability 99.9%, the core is among those t for which $\psi_0(t) \geq m - 4.5$.

3°. What about general case? The value (5.3.8) does not change if we re-scale all the parameters: $n(t) \to K \cdot n(t)$, $a_0 \to a_0/K$, $b_0 \to b_0/K$, $A_0 \to A_0/K$, and $B_0 \to B_0/K$, for any $K > 0$. W.l.o.g., we can therefore assume that $a_0 = 1$.

Differentiating (5.3.8) by a_0, we conclude that $a_0 = (\sum E(t))/(\sum n(t))$. Similarly, $b_0 = (\sum V(t))/(\sum n(t))$. Since $a_0 = 1$, we thus get $b_0 = (\sum V(t))/(\sum E(t))$. Differentiating by A_0, we conclude that

$$A_0 = \frac{1}{N} \sum_t \frac{(E(t) - n(t) \cdot a_0)^2}{n(t)} = \frac{1}{N} \left(\sum_t \frac{E(t)^2}{n(t)} - \sum_t E(t) \right) \tag{5.3.10}$$

and similarly,

$$B_0 = \frac{1}{N} \left(\sum_t \frac{V(t)^2}{n(t)} - b_0 \cdot \sum_t V(t) \right). \tag{5.3.11}$$

If we denote $\lambda \stackrel{\text{def}}{=} A_0/B_0$, then the above formula for $n(t)$ takes the form $n(t)^2 = (E(t)^2 + \lambda \cdot V^2(t))/(1 + \lambda \cdot b_0^2)$. Substituting this expression into (5.3.10) and (5.3.11) and using the

fact that $A_0 = \lambda \cdot B_0$, we conclude that

$$\sum_t \frac{E(t)^2}{\sqrt{E(t)^2 + \lambda \cdot V(t)^2}} \cdot \sqrt{1 + \lambda \cdot b_0^2} - \sum_t E(t) =$$

$$\sum_t \frac{\lambda \cdot V(t)^2}{\sqrt{E(t)^2 + \lambda \cdot V(t)^2}} \cdot \sqrt{1 + \lambda \cdot b_0^2} - b_0 \cdot \left(\sum_t V(t) \right)$$

with the only unknown λ. After we find λ from this equation, we can thus find A_0, B_0, and hence, the desired t_0.

To test our technique, we simulated an explosion with randomly distributed fragments. On this simulation, the above algorithm does detect the core.

Possibility of parallelization. In the above algorithms, processing values corresponding to bin i uses only measurement only from this bin and from the neighboring bins. Therefore, if we have several processors working in parallel (see, e.g., [87]), we can speed up the computations by having each processor process a section of bins. For example, for 2 processors, the first can handle bins 1 to $N/2+n$, and the second all the bins from $N/2-n$ to N, where n is the number of neighboring bins that we need to take into consideration.

Multiple explosions: case of a very accurate radar. Sometimes, the observed fragments cloud comes not from a single explosion, but from several consequent explosions. How can we then determine the core?

Let us show that when the radar is accurate enough, so that we can distinguish between individual fragments, the problem of determining the core becomes even easier than in the case of a single explosion.

First, we observe that if the radar is that accurate, then, by making observations at very close moments of time T_1, T_2, etc., we can *trace* individual fragments. Indeed, at the initial moment T_1, we identify fragments by the times $t_1^{(1)} < t_1^{(2)} < \ldots$ at which the corresponding signal $s_1(t)$ is non-zero. At the next moment T_2, we can find the times t_2, t_2', ... corresponding to the fragments as the times t for which $s_2(t) \neq 0$. When the time

difference $T_2 - T_1$ is so small that the relative motion of a fragment is smaller than the distance between different fragments, we can identify, for each fragment i, the corresponding time $t_2^{(i)}$ as the closest to $t_1^{(i)}$ among all observed values t_2, t_2', ...

For a single explosion, a linear formula (5.3.3) relates $t_2^{(i)}$ and $t_1^{(i)}$; the corresponding slope a_{kl} depends on the moment T_0 of the explosion. If two explosions occurred at moments T_0 and T_0', we get similar linear formulas for the fragments of each explosion, with two slopes $a_{kl} \neq a_{kl}'$. Thus, by plotting the dependence of $t_2^{(i)}$ on $t_1^{(i)}$, we will get two straight lines with different slopes. The core belongs to both families of fragments. Thus, the core can be determined as the fragment i_0 that lies at the intersection of the two corresponding straight lines.

For two explosions, we can determine both lines and easily find the intersection. For numerous explosions, we will have many straight lines, and finding all of them may be computationally difficult; so, we need a different idea.

The dependence of a_k on T_0 is monotonic, so in such situations, the 2-D points $t^{(i)} \stackrel{\text{def}}{=} (t_1^{(i)}, t_2^{(i)})$ occupy a zone between two straight lines with different slope $\underline{a} < \overline{a}$ corresponding to the first and the last explosions; geometrically, it is a 2-D cone with the core's value $t^{(i_0)}$ as the vertex. Since we have numerous explosions, we can conclude that the corresponding pairs fill the entire cone.

Let us show that the core can be determined as the only value i for which

$$\max_{j:\, t_1^{(j)} < t_1^{(i)}} t_j^{(2)} < \min_{j:\, t_1^{(j)} > t_1^{(i)}} t_j^{(2)}. \tag{5.3.12}$$

Let us first consider the case $i = i_0$. For each of the corresponding straight lines, the dependence of $t_2^{(i)}$ on $t_1^{(i)}$ is monotonically increasing; since the core i_0 belongs to all the lines, we can therefore conclude that if $t_1^{(j)} < t_1^{(i_0)}$, then we have $t_2^{(j)} < t_2^{(i_0)}$, and if $t_1^{(j)} > t_1^{(i_0)}$, then we have $t_2^{(j)} > t_2^{(i_0)}$ – which implies (5.3.12).

If $t_1^{(i)} > t_1^{(i_0)}$, then the maximum in the left side of the formula (5.3.12) corresponds to the largest possible slope $\overline{a}_{kl}$ and is therefore equal to $t_2^{(i_0)} + \overline{a}_{kl} \cdot (t_1^{(i)} - t_1^{(i_0)})$. On the other hand, the minimum in the right side of the formula (5.3.12) corresponds to the smallest

possible slope slope $\underline{a}_{kl}$ and is therefore equal to $t_2^{(i_0)} + \underline{a}_{kl} \cdot (t_1^{(i)} - t_1^{(i_0)})$ – which is clearly smaller than the maximum in the left side of (5.3.12).

Similarly, (5.3.12) cannot occur for $t_i^{(i)} < t_1^{(i_0)}$.

5.4 Image Processing

We analyze the problem of inverse half-toning. This problem is a particular case of a class of difficult-to-solve problems: inverse problems for reconstructing piece-wise smooth images. We show that this general problem is NP-hard. We also propose a new idea for solving problems of this type, including the inverse halftoning problem.

5.4.1 Introduction

Need for halftoning. Inside the computer, a gray-scale image is represented by assigning, to every pixel (n_1, n_2), the intensity $f(n_1, n_2)$ of the color at this pixel. Usually, 8 bits are used to store the intensity, so we have $2^8 = 256$ possible intensity levels for each pixel.

For color images, we must represent the intensity of each color component.

A laser printer cannot print the points of different intensity; at any pixel, it either prints a black (or a colored) dot, or it does not print anything at all. Therefore, when we print an image, we must first transform it into the form $b(n_1, n_2)$ in which at every pixel (n_1, n_2), we only have 0 or 1: 0 if we do not print a black dot at this location, and 1 if we do. This transformation from the original continuous image to the two-level ("halftone") image is called *halftoning*.

Crudely speaking, the level of intensity at a pixel is represented by the relative frequency of black spots around it:

- if the original image was black, then all the neighboring pixels are black;
- if the original image was white, then none the neighboring pixels are black, they are all white;

- any level between absolute black and absolute white means that some pixels in the neighborhood are black and some are white; the larger the intensity, the more black pixels there is.

Halftoning techniques: a brief reminder. There exist many halftoning algorithms; see, e.g., [207]. One of the most widely used halftoning algorithms is the *error diffusion* algorithm [207], in which we start with the original image $u(n_1, n_2) := f(n_1, n_2)$ and sequentially update the processed image $u(n_1, n_2)$ and quantize the processed value $u(n_1, n_2)$ into the halftone image $b(n_1, n_2) = Q(u(n_1, n_2))$, where:

- $Q(u) = 0$ for $u < 0.5$ and
- $Q(u) = 1$ for $u \geq 0.5$.

Once the pixel is quantized, the quantization error $e(n_1, n_2) \stackrel{\text{def}}{=} b(n_1, n_2) - u(n_1, n_2)$ is spread out ("diffused") to the values $u(n_1, n_2)$ at neighboring pixels, so that the processed value $u(n_1, n_2)$ eventually becomes equal to

$$u(n_1, n_2) = f(n_1, n_2) - \sum_{m_1, m_2} h(m_1, m_2) \cdot e(n_1 - m_1, n_2 - m_2).$$

Need for reverse halftoning. Visually, the halftone image printed on a high-quality laser printer looks practically identical to the original gray-scale image that we can see on the computer screen. So, visually, once we have a halftone image $b(n_1, n_2)$, we can tell which original multi-level image $f(n_1, n_2)$ it came from. However, this intuitive understanding is difficult to describe in precise terms.

Once we have a printed image, we can digitally scan it and get the halftone values $b(n_1, n_2)$ from this printed page. From these halftone values, we would like to reconstruct the original image. Our eyes can do it, but it is not so easy to describe this ability in algorithmic terms.

The need for such a representation also comes from the need to manipulate the original image, e.g., rotate it or zoom on it. These operations are easy to perform on the original

image, but it is not clear how to perform them on a halftone image. So, if we want to go from a printed image to a printed zoomed and/or rotated image, we can do it in this way:

- first, we use the halftone image $b(n_1, n_2)$ to reconstruct the original image $f(n_1, n_2)$;
- then, we apply the appropriate zoom and/or rotation operations to the reconstructed image $f(n_1, n_2)$, resulting in a transformed image $f^*(n_1, n_2)$;
- finally, we halftone the transformed image $f^*(n_1, n_2)$, and print the resulting halftone image $b^*(n_1, n_2)$.

In all these cases, we must reverse the halftoning procedure.

Halftoning is an ill-posed problem: a reminder. Our objective is to reverse the halftoning operation. By definition, halftoning transforms the original gray-scale image in which we stored at least 8 bits per pixel, into a black-and-white image in which we store only one bit per pixel. Thus, halftoning loses information and is, therefore, a lossy compression.

Hence, there may be several different images that lead to the same halftoned image.

Existing inverse halftoning techniques: POCS. There exist several different techniques for inverse halftoning:

One class of such techniques is based on the iterative projection onto convex set (POCS); see, e.g., [81, 132]. Crudely speaking, the main idea behind these methods is that each value $b(n_1, n_2)$ of a halftone image represents a constraint on the original image $f(n_1, n_2)$. In most halftoning methods like error diffusion halftoning, the relation between the original image $f(n_2, n_2)$ and the halftone image $b(n_1, n_2)$ is described by convex transformations, so for each value $b(n_1, n_2)$, the set of all the images that lead to this particular value is a convex set.

Thus, the set of all the images $f(n_1, n_2)$ which are consistent with the halftone image $b(n_1, n_2)$ is also a convex set. Among these images, we want to find an image that satisfies

certain reasonable properties, e.g., an image that is sufficiently smooth. For many properties, the set S of such images is convex. We would like to find, among all the images from the class S, the closest to $b(n_1, n_2)$ (e.g., in L^2 metric) that is consistent with the halftone image $b(n_1, n_2)$.

From the geometric viewpoint, we have a point $b(n_1, n_2)$ in the function space, and we want to find the closest element to this point in the convex set that is the intersection of the set S of all desirable images and the convex sets formed by all the images that lead to this very halftone image. It is known that to get this projection, we can:

- project our point onto the first of the intersected convex sets, then
- project the resulting point onto the second,
- etc.,

iterating the procedure if necessary. In terms of constraint propagation, we need to satisfy several constraints, so we:

- first minimally modify the original halftone image so that it satisfies the first constraint,
- then minimally modify the modification so that it satisfies the second constraint,
- etc.

The resulting projection on convex sets method indeed leads to a good quality inverse halftoning.

Existing inverse halftoning techniques: wavelet techniques. Another class of techniques for inverse halftoning uses wavelet transform, a techniques that, as the experience of JPEG2000 has shown (see, e.g., [198]), best captures the visual quality of images uncompressed after a lossy compression (and halftoning is, as we have mentioned, an example of lossy compression).

Wavelet-based techniques for inverse halftoning are presented, e.g., in [132]. In accordance with the JPEG2000 experience, wavelet-based inverse halftoning techniques lead to visually the best reconstruction among all known inverse halftoning methods.

Existing inverse halftoning techniques: fast techniques based on adaptive filtering. As we have just mentioned, the wavelet-based inverse halftoning techniques lead to visually the best reconstruction among all known inverse halftoning methods. The only reason why these methods are not universally used is that these methods take a lot of computations.

In view of this fact, researchers have been trying to design faster inverse halftoning techniques that would still lead to similar quality image reconstruction.

The main idea behind such techniques is that, as we have mentioned, the intensity of the original image $f(n_1, n_2)$ can be reconstructed from the density of black pixels in the neighborhood of a given pixel (n_1, n_2). In engineering terms, this means that the original image $f(n_1, n_2)$ can be obtained from the halftone image $b(n_1, n_2)$ by low-pass filtering.

If the image consists of a single object, with intensity smoothly changing from pixel to pixel, then we can indeed apply a low-pass filter to the half-tone image and get a reasonable reconstruction. However, in real life, images have edges. When applied to an image with edges, a low-pass filter correctly reconstructs the intensity within each smooth zone, but blurs the edge.

A natural way to avoid this blurring is to detect the edges and to apply different filters (with different spatial radius) at different parts of the image, so that:

- a filter applied inside each zone would have a larger radius and thus, have a greater smoothing effect, while
- a filter applied closer to the edge would have a smaller radius, and thus, would preserved the edge.

Of course, ideally, instead of just two levels inside-edge, we should have a filter radius

adjusted to the estimated gradient of intensity at the given pixel (n_1, n_2).

This idea has been successfully implemented in inverse halftoning; see, e.g., [98]. The resulting method is much faster than the wavelet-based reconstruction, while the visual quality of the reconstructed images is almost as good as for the wavelet-based reconstruction.

Inverse halftoning: remaining problem. The remaining problem is that the existing methods are still not optimal. They are optimized with respect to selection of parameters, but the consensus of researchers is that there is still room for improvement, especially when we are looking for methods of low computational intensity that can be easily implemented within the existing printing devices.

What we are planning to do? In this dissertation, we will show that the problem of inverse half-toning is a particular case of a class of difficult-to-solve problems: inverse problems for reconstructing piece-wise smooth images. We show that this general problem is NP-hard. We also propose a new idea for solving problems of this type, including the inverse halftoning problem.

5.4.2 Inverse Halftoning is a Particular Case of a General Class of Inverse Problems of Reconstructing Piecewise Smooth Images

Inverse halftoning is an ill-posed problem. We have mentioned that the inverse halftoning problem is ill-posed in the sense that, from the purely mathematical viewpoint, there are many different images that are consistent with the same observed data – in this case, with the given halftone image $b(n_1, n_2)$.

Most inverse problems in science and engineering are ill-posed. The above ill-posedness of the inverse halftoning problem is a common feature in applications: most

inverse problems in science and engineering are ill-posed; see, e.g., [199].

Smoothness: traditional approach to solving ill-posed inverse problem. A typical way to solve an inverse problem is to find a natural physically meaningful property of actual solution, and use this *a priori* information to select a single most physically meaningful solution among many mathematically possible ones. This process is called *regularization.*

Typically, in inverse problems, this natural property is smoothness. Smoothness can be naturally described in precise mathematical terms. For example, when we reconstruct a 1-D signal $x(t)$, then the degree of smoothness can be defined as follows. At a given moment of time t, the larger the absolute value $|x'(t)|$ of the derivative $x'(t)$, the less smooth the signal is. Thus, at a given time t, the value $|x'(t)|$ is a natural degree of the signal's non-smoothness. Overall, a natural degree of non-smoothness can be defined as a mean square of these degrees corresponding to different moments t, i.e., as $J \stackrel{\text{def}}{=} \int (x'(t))^2\,dt$.

Most regularization techniques try to find, among many signals that are consistent with given observations, the smoothest signal, i.e., the signal with the smallest possible value of the degree of non-smoothness J.

Smoothness: discrete case. In real life, we only have the values $x(t_1)$, $x(t_2)$, ..., of the signal $x(t)$ at discrete moment of time t_1, $t_2 = t_1 + \Delta t$, ..., $t_{i+1} = t_i + \Delta t$, ... Based on this discrete data, we can approximate the derivative $x'(t)$ as a difference $\dfrac{x(t_{i+1}) - x(t_i)}{\Delta t}$, so minimizing the integral J is equivalent to minimizing the corresponding integral sum $J_{\text{discr}} \stackrel{\text{def}}{=} \sum\limits_i (x(t_{i+1}) - x(t_i))^2$.

Smoothness: 2-D case. For a 2-D image $f(n_1, n_2)$, similarly, a natural assumption is that this image is smooth. Similarly to the 1-D case, a natural way to describe the degree of smoothness of a given image is to use the integral sum

$$J \stackrel{\text{def}}{=} \sum_{n_1, n_2} [(f(n_1+1, n_2) - f(n_1, n_2))^2 + (f(n_1, n_2+1) - f(n_1, n_2))^2].$$

Alternatively, we can describe this criterion as the sum of the squares of the differences in intensity between all possible pairs (p, p') of neighboring pixels $p = (n_1, n_2)$ and $p' = (n_1', n_2')$:

$$J = \sum_{p,p' \text{ are neighbors}} (f(p) - f(p'))^2.$$

Smoothness makes problems computationally solvable. A practically useful property of the above degrees of non-smoothness J is that the expression J is a convex function of the signal $x(t_i)$ or $f(n_1, n_2)$. Thus, if the conditions describing the fact that the unknown images is consistent with the observations is also described by linear or, more generally, smooth inequalities, then the problem of finding the regularized solution can be reformulated as a problem of minimizing a convex function J on the convex set.

Similarly, if we fix the degree of non-smoothness and look, among all the solutions with a given degree of non-smoothness, for the one that is the closest to the original approximate solution, we also have a problem of minimizing a convex function (distance) on the convex set (of all functions that are consistent with the observations and have the desired degree of smoothness).

It is known that, in general, the problems of minimizing convex functions over convex domains are algorithmically solvable (see, e.g., [201]), and smoothness-based regularization has indeed been efficiently implemented; see, e.g., [199].

For image reconstruction, we only have piecewise smoothness. We have already mentioned that in images, we have a smooth change from pixel to pixel only within an object; between objects, we may have a sharp edge in which there is no smoothness at all.

Many other inverse problems can also be characterized by piecewise smoothness. For example, in the inverse problem of geophysics, we use the results of ultrasound waves passing through the earth to find out how the density change with depth and location. In geophysics, we have clear layers of different rocks with sharp edges between different layers, so we also face an inverse problem with only piecewise smoothness; see, e.g., [171].

Traditional smoothness measures are not adequate for piecewise smoothness. In the piecewise smooth case, the above measure of non-smoothness is not applicable, because it would include neighboring pixels on the different sides of the edge.

Appropriate smoothness measures for piecewise smoothness case. To avoid the above problem, we need to only take into account the pairs of neighboring pixels that belong to the same zone, i.e., consider the sum

$$J(Z) = \sum_{p,p' \text{ are neighbors in the same zone}} (f(p) - f(p'))^2,$$

where Z denotes the information about the zones. This measure makes computational sense only if we know beforehand where the zones are – i.e., where is the border between the two zones.

However, in real life, finding the edge is a part of the problem. In this case, we can use the same smoothness criterion not only to reconstruct the original image, but also to find the edge location. Specifically, we want to look for the zone distribution *and* for the zone location for which the above criterion J takes the smallest possible value.

In terms of an image, we fix the number of zones, and we characterize the non-smoothness of an image by a criterion

$$J^* = \min_{\text{all possible divisions } Z \text{ into zones}} J(Z).$$

The resulting problem is no longer convex. The resulting functional is no longer convex, because the division into zones is a discrete problem. It is known that non-convex problems are, in general, more computationally difficult than the corresponding convex ones (see, e.g., [95]), and adding discrete variables makes the problems even more computationally difficult; see, e.g., [170].

5.4.3 New Complexity Result: Complexity of Inverse Problems of Reconstructing Piecewise Smooth Images

In this section, we show that in general, the inverse problem for piecewise smooth case is computationally intractable (NP-hard) even when the relation expressing the consistency between the measured results and the desired image is linear.

This proof will follow the proof of NP-hardness of different image and signal processing problems described in our previous publications [110].

Let us prove that in general, the inverse problem for piecewise smooth case is computationally intractable (NP-hard).

Main idea of the proof: reduction to a subset problem. To prove NP-hardness of our problem, we will reduce a known NP-hard problem to the problem whose NP-hardness we try to prove: namely, to the inverse problem for piecewise smooth images.

Specifically, we will reduce, to our problem, the following *subset sum* problem [110, 170] that is known to be NP-hard:

- Given:
  - m positive integers $s_1, \ldots, s_m$ and
  - an integer $s > 0$,
- check whether it is possible to find a subset of this set of integers whose sum is equal to exactly s.

For each i, we can take $x_i = 0$ if we do not include the i-th integer in the subset, and $x_i = 1$ if we do. Then the subset problem takes the following form: check whether there exist values $x_i \in \{0, 1\}$ for which $\sum s_i \cdot x_i = s$.

We will reduce each instance of this problem to the corresponding piecewise smooth inverse problem.

Reduction to a subset problem: details. Let us consider the following problem. We want to reconstruct an $m \times m$ image $f(n_1, n_2)$. Let $d = \lfloor m/2 \rfloor$. We want a piecewise smooth image $f(n_1, n_2)$ that consists of two zones.

The following linear constraints describe the consistency between the observations and the desired image:

- $f(n_1, n_2) = 1$ for $n_2 > d$;
- $\sum\limits_{i=1}^{m} s_i \cdot f(i, d) = s$; and
- $f(n_1, n_2) = 0$ for $n_2 < d$.

The problem that we consider is to find the solution with the smallest possible value of smoothness J^* among all the images that satisfy these linear constraints.

Let us show that the minimum of J^* is 0 if and only if the original instance of the subset problem has a solution. Indeed, if J^* is 0, this means that all the values within each zone must be the same. Since we have values 1 for $n_2 > d$ and values 0 for $n_2 < d$, we must therefore have every value to be equal either to 0 or to 1. Thus, if we have such a solution, the corresponding values $f(i, d) \in \{0, 1\}$ provide the solution to the original subset problem $\sum s_i \cdot x_i = s$.

Vice versa, if the selected instance of the original subset problem has a solution x_i, then we can take $f(i, d) = x_i$ and get the solution of the inverse problem for which the degree of non-smoothness is exactly 0.

So, if we can solve the inverse problem for piecewise smooth images, we will thus be able to solve the subset sum problem.

This reduction proves that the inverse problem for piecewise smooth images is indeed NP-hard.

5.4.4 Towards Possible Use of Interval Computations in Inverse Halftoning

Every halftoning algorithm includes a thresholding step. For every halftoning algorithm, including the error diffusion algorithm, there is a thresholding step where we replace the original continuous value with the quantized value. Usually, to decide whether the halftoned value $b(n_1, n_2)$ at a pixel (n_1, n_2) will be 0 (white pixel) or 1 (black pixel), we do some processing on the original image $f(n_1, n_2)$, and then apply thresholding to the resulting value $u(n_1, n_2)$. For example, in the error diffusion halftoning, we compute the auxiliary value $u(n_1, n_2)$, and then compute the halftone as $b(n_1, n_2) = Q(u(n_1, n_2))$, where:

- $Q(u) = 0$ if $u < 0.5$ and
- $Q(u) = 1$ if $u \geq 0.5$.

New idea: instead of selecting a single original image, produce the whole range of possible original images. We have already mentioned that halftoning loses information and is, therefore, a lossy compression. Hence, there may be several different images that lead to the same halftoned image.

In the existing methods, we reverse the halftoning procedure by selecting one of such images. However, it may be beneficial to present not just a *single* possible original image, but the whole *range* of images that could lead to the given halftoned image.

For example, in the above halftoning procedure, if we know that $b(n_1, n_2) = 0$, this means that the signal $u(n_1, n_2)$ could have any value from the interval $(0.0, 0.5)$, while if we know that $b(n_1, n_2) = 1$, this means that the signal $u(n_1, n_2)$ could have any value from the interval $[0.5, 1.0)$.

Result: interval-valued image. If we follow this idea, then instead of the image in which the intensity $f(n_1, n_2)$ at every pixel has an exact value, we come up with

an "interval-valued" image in which, at each pixel (n_1, n_2), we only know the interval $[\underline{f}(n_1, n_2), \overline{f}(n_1, n_2)]$ of possible values of intensity.

A similar idea of interval-valued quantities has been successfully used in science and engineering. The idea of using interval-valued quantities to represent uncertainty in engineering and scientific applications is not new: it has been known since the 1950s when R. E. Moore from Lockheed used it to analyze trajectories of intercontinental missiles and spaceship under interval uncertainty; see, e.g., [136]. Since then, interval computations have been widely applied in different engineering problems; see, e.g., [86, 88, 94, 145].

Interval techniques have been actively used in robust control [88], and in image and data processing [28, 102]. We believe that interval techniques can be applied to the inversion of halftoning as well.

Interval methods may provide one more explanation for the efficiency of wavelets. In particular, there exists an interval-based justification of wavelet techniques in image processing [28].

This explanation makes us believe that interval methods may be able to explain why wavelet-based inverse halftoning techniques lead, at present, to the most accurate (albeit not the fastest) inverse halftoning.

5.4.5 New Algorithm for Interval-Motivated Inverse Halftoning

As we have mentioned, low-pass filtering of the halftone (binary) image $b(n_1, n_2)$ provides a good first approximation $\ell(n_1, n_2)$ to the original image. However, the resulting lowpass filtered image is usually still different from the original image $f(n_1, n_2)$: $\ell(n_1, n_2) \neq f(n_1, n_2)$.

One reason for this difference is that, as we have mentioned, halftoning is a lossy compression. Due to this lossiness, several different images $f(n_1, n_2)$ lead to the same halftone image $b(n_1, n_2)$, so we cannot reconstruct the original image exactly.

However, there is another reason why $\ell(n_1, n_2) \neq f(n_1, n_2)$: namely, when we apply

the original halftoning to the result $\ell(n_1, n_2)$ of applying the low-pass filter to the halftone image $b(n_1, n_2)$, we do not exactly get back the same halftone image. In this sense, the lowpass filtered image is not the true inverse to the halftoning procedure.

It is therefore desirable to modify the lowpass filtered image so that the modified image will be inverse to halftoning, in the sense that if we apply the halftoning procedure to the modified image $g(n_1, n_2)$, we will get exactly the halftone image $b(n_1, n_2)$.

An interval consistency algorithm: description. Let us show how we can use interval ideas to design a desired image modification procedure. We will apply these ideas to the most widely used halftoning algorithm – error diffusion. In error diffusion, in order to process a pixel (n_1, n_2), we must have the results of halftoning of pixels $(n_1 - m_1, n_2 - m_2)$ with smaller values of the coordinates. Thus, in this halftoning procedure, we start processing the image with the pixel (1,1), and then we proceed with pixels (n_1, n_2) with increasing values of n_1 and n_2.

To invert the halftone image, we similarly start with the pixel (1,1). The result $b(1, 1)$ of halftoning this pixel depends only on the intensity $f(n_1, n_2)$ at this pixel: $b(n_1, n_2) = 0$ for $f(n_1, n_2) < 0.5$ and $b(n_1, n_2) = 1$ for $f(n_1, n_2) \geq 0.5$. So, to check whether halftoning of $\ell(n_1, n_2)$ produces the correct value of $b(1, 1)$, it is sufficient to apply the above thresholding to the value $\ell(1, 1)$. If the result of this thresholding coincides with $b(1, 1)$, we keep the lowpass filtered value $\ell(1, 1)$, i.e., we set $g(1, 1) = \ell(1, 1)$.

On the other hand, if the result of thresholding $\ell(1, 1)$ is different from the halftone value $b(1, 1)$, then we would prefer to select $g(1, 1)$ from the corresponding interval $(0.0, 0.5)$ or $[0.5, 1.0)$ of values that lead to the correct $b(1, 1)$. As we have mentioned, ideally, we should keep the entire interval as an interval of possible values of the original image; however, in this first algorithm, we select a single point $g(1, 1)$ within this interval – the point which is the closest to the lowpass filtered value $\ell(1, 1)$.

In other words:

- if $b(1, 1) = 0$ and $\ell(n_1, n_2) \geq 0.5$, then we take $g(n_1, n_2) = 0.5 - \varepsilon$ (where ε is a small

positive number, e.g., the smallest positive floating point number representable in the given computer), and

- if $b(1,1) = 1$ and $\ell(n_1, n_2) < 0.5$, then we take $g(n_1, n_2) = 0.5$.

After producing $g(1,1)$, we proceed to the next pixel, etc. Once we get to the pixel (n_1, n_2), this means that we have already processed the previous pixels. This means that we have already produced the values $g(n_1', n_2')$ for all the coordinates $n_1' < n_1$ and $n_2' < n_2$, and, correspondingly, the values $u(n_1', n_2')$ and $e(n_1', n_2')$ (see the description of error diffusion halftoning in Section I).

We want to select $g(n_1, n_2)$ at the pixel (n_1, n_2) in such a way that:

- first, the result of halftoning $g(n_1, n_2)$ is exactly the value $b(n_1, n_2)$;
- second, if there are several such values $g(n_1, n_2)$, then among these values, we would like to select the value that is the closest to the lowpass filtered image $\ell(n_1, n_2)$.

As we have mentioned in Section I, the value $b(n_1, n_2)$ of halftoning $g(n_1, n_2)$ is the result of thresholding the linear combination $u(n_1, n_2) = g(n_1, n_2) - g_0(n_1, n_2)$, where

$$g_0(n_1, n_2) \stackrel{\text{def}}{=} \sum_{m_1, m_2} h(m_1, m_2) \cdot e(n_1 - m_1, n_2 - m_2).$$

So, if $g(n_1, n_2) = \ell(n_1, n_2)$ leads to the correct halftoning, i.e., if the thresholding of $u(n_1, n_2) = \ell(n_1, n_2) - g_0(n_1, n_2)$ leads to the desired value $b(n_1, n_2)$, then we select $g(n_1, n_2) = \ell(n_1, n_2)$.

On the other hand, if the result of thresholding $g(n_1, n_2) = \ell(n_1, n_2) + g_0(n_1, n_2)$ is different from $b(n_1, n_2)$, then we take, as $g(n_1, n_2)$, the closest value from the corresponding interval.

When $b(n_1, n_2) = 1$, then the corresponding interval for $g(n_1, n_2) + g_0(n_1, n_2)$ is $[0.5, 1.0)$, so the interval of desired values of $g(n_1, n_2)$ is $[0.5 - g_0(n_1, n_2), 1.0)$. Thus, if the lowpass filtered value $\ell(n_1, n_2)$ is not in this interval, we select, as $g(n_1, n_2)$, the closest value from this interval, i.e., $g(n_1, n_2) = 0.5 + g_0(n_1, n_2)$.

Similarly, when $b(n_1, n_2) = 0$ and $\ell(n_1, n_2)$ does not belong to the corresponding interval $(0.0, 0.5 - g_0(n_1, n_2))$, we select, as $g(n_1, n_2)$, the closest value from this interval, i.e., $g(n_1, n_2) = 0.5 - g_0(n_1, n_2) - \varepsilon$.

A POCS iterative procedure. The interval consistency algorithm just described is evaluated in a POCS iterative procedure so that its impact can be evaluated using the Floyd-Steinberg algorithm for error diffusion based halftoning. The input halftoned image is first lowpass filtered using a 5x5 Gaussian lowpass mask. The low pass filtering removes the high frequency halftoning noise as well as all other high frequency information available in the halftoned original. The image as a result of lowpass filtering is not only blurred but is no longer a valid candidate image. That is, the difference between the original halftoned image and the rehalftoned version is not zero. The lowpass filtered image is then processed by the interval consistency algorithm that will make it a candidate image. This limiting step induces some large local differences in gray level which can be fused back into the image by a projection step that replaces the low-part of the frequency spectrum with that of the halftone input image. This step is called frequency swapping which is implemented using the two-dimensional DFT. The limiting and frequency swapping steps are then repeated a few times to produce the final output image.

Summary. The related disadvantage of our algorithm is that, while it produces an image $g(n_1, n_2)$ whose halftoning produces the exact same result as the original image $f(n_1, n_2)$, for standard benchmark images $f(n_1, n_2)$, the visual difference between $g(n_1, n_2)$ and $f(n_1, n_2)$ is higher than, e.g., for methods from [98].

Chapter 6

Conclusions and Future Work

6.1 Formulation of the Problem: Reminder

Statistical analysis and, more generally, statistical data processing is an extremely important part of modern science and engineering. It is therefore necessary to use the most adequate statistical techniques. Many traditional statistical formulas $y = C(x_1, \ldots, x_n)$, e.g., the formulas for the population mean, population variance, etc., are based on the simplifying assumptions that for some sample, we know the exact values $x_1, \ldots, x_n$ of the quantity of interest. In reality, these values usually come from measurements or from expert estimates; both methods are not absolutely accurate. It is thus desirable to find out how the inaccuracy with which we know the sample values affects the result of statistical analysis.

In many real-life situations, we only know the upper bound Δ on the measurement inaccuracy. In this case, once we know the result $\widetilde{x}$ of measuring the desired quantity, the only information that we have about the actual (unknown) value x of this quantity is that this value must belong to the interval $[\widetilde{x} - \Delta, \widetilde{x} + \Delta]$. Thus, it is necessary to generalize traditional statistical formulas $C(x_1, \ldots, x_n)$ to the case when we only know the inputs with interval uncertainty. In precise terms, we know n intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and we would like to compute the range $\mathbf{C}$ of possible values of the given characteristic $C(x_1, \ldots, x_n)$ when $x_i \in \mathbf{x}_i$.

In general, this computational problem is NP-hard even for the variance. However, in some practically important situations, it has been possible to design efficient algorithms for computing these characteristics. The main objective of this research was to improve

the existing algorithms and to design new algorithms for practically important situations which were not covered by the previous algorithms.

6.2 Main Results of this Dissertation

In this dissertation, we reduce the computational complexity of several known efficient algorithms to the complexity $O(n \cdot \log(n))$ comparable with the complexity of simply sorting the given values, and in some situations, even to the linear time complexity $O(n)$ – the smallest possible computational complexity of any algorithm for processing n input values $x_1, \ldots, x_n$. We also designed new efficient algorithms for computing variance and other important statistical characteristics for several practically important classes of statistical problems.

Specifically, for the lower endpoint $\underline{V}$ for the variance V, the fastest previously known algorithm takes $O(n \cdot \log(n))$ time. In the dissertation, we developed a linear time algorithm for computing $\underline{V}$.

The problem of computing the upper endpoint $\overline{V}$ is, in general, NP-hard. Previously, $O(n^2)$ time algorithms were known for the cases of narrow intervals (when no two intervals intersect), of slightly wider intervals (when for some K, no group of K intervals has a common point), and of privacy (when every two intervals either coincide or do not intersect). In the dissertation, we reduced the computational complexity of computing $\overline{V}$ to linear time for narrow intervals and for the privacy case, and to $O(n \cdot \log(n))$ times for slightly wider intervals. We also designed a new linear time algorithm for the case of a single measuring instrument MI (when no interval is a proper subset of another one), and an $O(n^m)$ time algorithm for the case of m measuring instruments, when input intervals can be divided into m classes each of which satisfies the above single-MI no-subset property.

We produced similar results for estimating the range of the endpoints $L = E - k_0 \cdot \sigma$ and $U = E + k_0 \cdot \sigma$ of the confidence interval which is often used to detect outliers (with $k_0 = 2$, $k_0 = 3$, or $k_0 = 6$). For the cases of narrow and slightly wider intervals, we reduced

the computation time from $O(n^2)$ to $O(n \cdot \log(n))$. We also designed a new $O(n \cdot \log(n))$ time algorithm for the single MI case and a new $O(n^m)$ time algorithm for the case of m MI.

We also describe efficient algorithms for computing the range of another important statistical characteristics – skewness (third central moment), a characteristic important for describing possible asymmetry of the probability distributions.

For all these algorithms, their correctness has been mathematically proven.

Our algorithms were motivated by (and applied to) several applied problems. The corresponding applications are also presented in this dissertation.

In *computer security*, we deal with an interval method of preserving privacy in which for each sensitive field of data, instead of the actual values of the corresponding quantity, we only keep a range (interval) of possible values of this quantity: e.g., instead of the exact age of 28, we only keep a range $[20, 30]$. For this application, we have developed new efficient algorithms for estimating values of different statistical characteristics under such privacy-related interval uncertainty.

In *geosciences*, we deal with the inverse problem of geophysics, where we measure the seismic signals generated by artificial small explosions, and then reconstruct the velocities at different 3-dimensional points from the travel times of this seismic signal. In this problem, the existing algorithms often produce the velocities v which are outside the geophysically known intervals $[\underline{v}, \overline{v}]$ of possible values. In this dissertation, we show how to take this interval information into account when solving the inverse problems.

In *computer engineering*, we deal with the problem of estimating the clock cycle in chip design. Traditional methods for clock cycle estimation only consider interval (worst-case) uncertainty. In reality, we often also have additional information about the mean values of the delays in the corresponding gates and wires. In the dissertation, we show how this information can be used to provide more realistic estimates for the clock cycle and thus, improve the efficiency of the chips.

Finally, in 1-dimensional *radar data processing*, we show how we can combine the prob-

abilistic uncertainty corresponding to measurement errors and interval uncertainty corresponding to finite distance resolution and thus distinguish the explosion core from the explosion fragments.

6.3 Remaining Open Problems

In theoretical and practical results presented in this dissertation, we solved the main algorithmic challenges formulated in the previous papers on statistical analysis under interval uncertainty. However, from the practical viewpoint, there are still many interesting and challenging open problems.

The first class of open problems comes from the fact that while we did decrease the computational complexity of many algorithms to a reasonable $O(n \cdot \log(n))$ time or even linear time, for some applications, we may need a further speed-up. Specifically, this speed-up is important in applications like real-time control where all the computations must be completed by the desired time, or in systems like "smart dust" microsensors which have very small computational capabilities. In some cases, the speed-up is necessary even for the regular computers. For example, for the case of m MI, our algorithms require $O(n^m)$ time. Formally, such algorithms are polynomial time and thus, they are usually considered feasible. However, when we use 10 different MI (a very realistic situation), we thus need $\approx n^{10}$ computation steps. Even for a small database of $n \approx 1000$ measurement results, this lead to an impractical amount of $(10^3)^{10} = 10^{30}$ computation steps.

The second class of open problems is related to the fact that we mainly concentrated on computing standard statistical characteristics such as mean and variance. In some practical applications, it is also important to know the values of other characteristics such as covariance, correlation, higher-order central moments, etc. In this dissertation, we provide estimates for one such rather rarely used characteristic – skewness, but there are many other characteristics for which interval bounds still need to be developed.

The third class of problems is related to the fact that in this dissertation, we mainly

concentrated on the case when we know all the values x_i with interval uncertainty. In reality, in addition to the upper bounds Δ_i on the corresponding measurement errors, we often also have some additional information about the measurement errors: e.g., the values which are most probably possible (case of fuzzy uncertainty), or partial information about the corresponding probabilities (case of probabilistic uncertainty).

These cases naturally appear in practical situations. In this dissertation, we covered several such cases from geosciences and chip design when we have to combine (fuse) different types of uncertainty. For example, in our applications to chip design, we covered the situation when, in addition to an interval of possible values of each parameter x_i, we also know the mean value of each of these parameters. For each specific case that we considered, we developed a *specific* algorithm. However, it is still a challenge to develop *general* ways of fusing different types of uncertainty.

References

[1] M. Ainsworth, and B. Senior, *Aspects of an hp-Adaptive Finite Element Method: Adaptive Strategy, Conforming Approximation and Efficient Solvers*, University of Leicester, England, U.K., Department of Mathematics and Computer Science, Technical Report 1997/2, 1997.

[2] U. O. Akpan, T. S. Koko, I. R. Orisamolu, and B. K. Gallant, Practical Fuzzy Finite Element Analysis of Structures, *Finite Elem. Anal. Des.*, 2001, Vol. 38, pp. 93–111.

[3] A. P. Annan, *Ground Penetrating Radar Workshop Notes*, Sensors and Software Inc., Mississauga, Ontario, Canada, 1992.

[4] Atlantis Scientific, Inc., *Theory of Synthetic Aperture Radar*, http://www.geo.unizh.ch/fpaul/sar-theory.html, 2005.

[5] R. Alo, M. Beheshti, and G. Xiang, Computing Variance Under Interval Uncertainty: A New Algorithm and Its Potential Application to Privacy in Statistical Databases, *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'06*, Paris, France, Jul. 2–7, 2006, pp. 810–816.

[6] S. W. M. Au-Yeung, *Finding Probability Distributions from Moments*, Master's Thesis, Imperial College, London, 2003.

[7] M. G. Averill, G. Xiang, V. Kreinovich, G. R. Keller, S. A. Starks, P. S. Debroux, and J. Boehm, How to Reconstruct the Original Shape of a Radar Signal?, *Proceedings of the 24th International Conference of the North American Fuzzy Information Processing Society NAFIPS'2005*, Ann Arbor, Michigan, Jun. 22–25, 2005, pp. 717–721.

[8] M. G. Averill, K. C. Miller, G. R. Keller, V. Kreinovich, R. Araiza, S. A. Starks, Using Expert Knowledge in Solving the Seismic Inverse Problem. International Journal of Approximate Reasoning, 2007, Vol. 45, No. 3, pp. 564–587.

[9] I. Babuška, M. Griebel and J. Pitkäranta, The Problem of Selecting the Shape Functions for p-Type Elements, *Int. J. Numer. Meth. Engrg.*, 1999, Vol. 28, pp. 1891–1908.

[10] I. Babuška, and W. Gui, The h, p, and hp-Versions of the Finite Element Method in One Dimension. Parts I–III, *Numer. Math.*, 1986, Vol. 49, pp. 577–683.

[11] I. Babuška, and T. J. Oden, Verification and Validation in Computational Engineering and Science: Basic Concepts, *Computer Methods in Applied Mechanics and Engineering*, 2004, Vol. 193, pp. 4057–4066.

[12] B. R. Barmish, *New Tools for Robustness of Linear Systems*, McMillan, New York, 1994.

[13] J. Beck, V. Kreinovich, and B. Wu, Interval-Valued and Fuzzy-Valued Random Variables: From Computing Sample Variances to Computing Sample Covariances, In: M. Lopez, M. A. Gil, P. Grzegorzewski, O. Hrynewicz, and J. Lawry, editor, *Soft Methodology and Random Information Systems*, Springer-Verlag, Berlin-Heidelberg, 2004, pp. 85–92.

[14] G. Bardossy, J. Fodor, *Evaluation of Uncertainties and Risks in Geology*, Springer Verlag, Berlin, 2004.

[15] M. Beheshti, J. Han, L. Longpré, S. A. Starks, J. I. Vargas, and G. Xiang, Interval and Fuzzy Techniques in Business-Related Computer Security: Intrusion Detection, Privacy Protection, *Proceedings of the Second International Conference on Fuzzy Sets and Soft Computing in Economics and Finance FSSCEF'2006*, St. Petersburg, Russia, Jun. 28 – Jul. 1, 2006, pp. 23–30.

[16] T. Belytschko, N. Moës, S. Usui, and C. Parimi, Arbitrary Discontinuities in Finite Elements, *International Journal of Numerical Methods in Engineering*, 2001, Vol. 50, No. 4, pp. 993–1013.

[17] Y. Ben-Haim, and I. Elishakoff, *Convex Models of Uncertainty in Applied Mechanics*, Elsevier Science, Amsterdam, 1990.

[18] D. Berleant, Automatically Verified Arithmetic with Both Intervals and Probability Density Functions, *Interval Computations*, 1993, No. 2, pp. 48–70.

[19] D. Berleant, Automatically Verified Arithmetic on Probability Distributions and Intervals, In: R. B. Kearfott and V. Kreinovich, editors, *Applications of Interval Computations*, Kluwer, Dordrecht, 1996.

[20] D. Berleant and C. Goodman-Strauss, Bounding the Results of Arithmetic Operations on Random Variables of Unknown Dependency Using Intervals, *Reliable Computing*, 1998, Vol. 4, No. 2, pp. 147–165.

[21] D. Berleant, M.-P. Cheong, C. Chu, Y. Guan, A. Kamal, G. Sheblé, S. Ferson, and J. F. Peters, Dependable Handling of Uncertainty, *Reliable Computing*, 2003, Vol. 9, No. 6, pp. 407–418.

[22] D. Berleant, L. Xie, and J. Zhang, Statool: A Tool for Distribution Envelope Determination (DEnv), an Interval-Based Algorithm for Arithmetic on Random Variables, *Reliable Computing*, 2003, Vol. 9, No. 2, pp. 91–108.

[23] D. Berleant and J. Zhang, Using Pearson Correlation to Improve Envelopes Around the Distributions of Functions, *Reliable Computing*, 2004, Vol. 10, No. 2, pp. 139–161.

[24] D. Berleant and J. Zhang, Representation and Problem Solving with the Distribution Envelope Determination (DEnv) Method, *Reliability Engineering and System Safety*, Jul.–Sep. 2004, Vol. 85, No.1–3.

[25] S. P. Bhattacharyya, H. Chapellat, and L. Keel, *Robust Control: The Parametric Approach*, Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[26] D. Boning and S. Nassif, Models of Process Variations in Device and Interconnect, In: A. Chandrakasan (ed.), *Design of High-Performance Microcomputer Circuits*, 2000.

[27] G. Bozkurt-Unal and A. E. Cetin, Restoration of Error-Diffused images using Projection onto Convex Sets, *IEEE Trans. Image Processing*, Dec. 2001, Vol. 10, No. 12, pp. 1836–1841.

[28] A. E. Brito and O. Kosheleva, Interval + Image = Wavelet: For Image Processing under Interval Uncertainty, Wavelets are Optimal, *Reliable Computing*, 1998, Vol. 4, No. 3, pp. 291–301.

[29] S. D. Cabrera, K. Iyer, G. Xiang, and V. Kreinovich, On Inverse Halftoning: Computational Complexity and Interval Computations, *Proceedings of the 39th Conference on Information Sciences and Systems CISS'2005*, John Hopkins University, Mar. 16–18, 2005, Paper 164.

[30] M. Ceberio, S. Ferson, V. Kreinovich, S. Chopra, G. Xiang, A. Murguia, and J. Santillan, How to Take Into Account Dependence Between the Inputs: From Interval Computations to Constraint-Related Set Computations, With Potential Applications to Nuclear Safety, Bio- and Geosciences, *Proceedings of the Second International Workshop on Reliable Engineering Computing*, Savannah, Georgia, Feb. 22–24, 2006, pp. 127–154.

[31] H. Chen, and T. Belytschko, An Enriched Finite Element Method for Elastodynamic Crack Propagation, *International Journal of Numerical Methods in Engineering*, 2003, Vol. 58, No. 12, pp. 1873–1905.

[32] S. H. Chen, H. D. Lian, and X. W. Yang, Interval Static Displacement Analysis for

Structures with Interval Parameters, *Int. J. Numer. Methods Engrg.*, 2002, Vol. 53, pp. 393–407.

[33] J. Chessa, and T. Belytschko, An Extended Finite Element Method for Two-Phase Fluids, *J. Appl. Mech.*, 2003, Vol. 70, No. 1, pp. 10–17.

[34] J. Chessa, and T. Belytschko, A Local Space-Time Discontinuous Finite Element Method, *Computer Methods in Applied Mechanics and Engineering* (to appear).

[35] J. Chessa, P. Smolinski, and T. Belytschko, The Extended Finite Element Method (X-FEM) for Solidification Problems, *Int. J. Numer. Methods Engrg.*, 2002, Vol. 53, pp. 1959–1977.

[36] J. Chessa, H. Wang, and T. Belytschko, On the Construction of Blending Elements for Local Partition of Unity Enriched Finite Elements, *Int. J. Numer. Methods Engrg.*, 2003, Vol. 57, No. 7, pp. 1015–1038.

[37] D. Chinnery and K. Keutzer, Closing the Gap Between ASICs and Custom, *Proc. of the Design Automation Conference DAC'2000.*

[38] D. Chinnery and K. Keutzer (eds.), *Closing the Gap Between ASICs and Custom*, Kluwer, 2002.

[39] P. G. Ciarlet, Discrete Maximum Principle for Finite Difference Operators, *Aequationes Math.*, 1970, Vol. 4, pp. 338–352.

[40] P. G. Ciarlet and P. A. Raviart, Maximum Principle and Uniform Convergence for the Finite Element Method, *Computer Methods in Applied Mechanics and Engineering*, 1973, Vol. 2, pp. 17–31.

[41] G. Corliss, C. Foley, and R. B. Kearfott, Formulation for reliable analysis of structural frames, In: R. L. Muhanna and R. L. Mullen (eds.), *Proc.*

NSF Workshop on Reliable Engineering Computing, Savannah, Georgia, 2004, http://www.gtsav.gatech.edu/rec/recworkshop/index.html

[42] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.

[43] N. Damera-Venkata, T. D. Kite, M. Venkataraman, and B. L. Evans, Fast Blind Inverse Halftoning, *Proc. IEEE Int. Conf. on Image Processing*, Oct. 4–7, 1998, Chicago, IL, vol. 2, pp. 64–68.

[44] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans, and A. C. Bovik, Image Quality Assessment Based on a Degradation Model, *IEEE Transactions on Image Processing*, Apr. 2000, Vol. 9, No. 4, pp. 636–650.

[45] J. J. Daniels, Fundamentals of Ground Penetrating Radar, *Proceedings of the Symposium on the Application of Geophysics to Engineering and Environmental Problems*, Colorado School of Mines, Golden, Colorado, 1989, pp. 62-142.

[46] J. J. Daniels, J. Bower, and F. Baumgartner, High Resolution Gpr at Brookhaven National Lab To Delineate Complex Subsurface Structure, *Journal Environmental and Engineering Geophysics*, 1989, Vol. 3, No. 1, pp. 1–6.

[47] J. J. Daniels, D. L. Grumman, and M. Vendl, Coincident antenna three dimensional GPR, *Journal of Environmental and Engineering Geophysics*, 1997, Vol. 2.

[48] E. Dantsin, V. Kreinovich, A. Wolpert, and G. Xiang, Population Variance Under Interval Uncertainty: A New Algorithm, *Reliable Computing*, 2006, Vol. 12, No. 4, pp. 273–280.

[49] E. Dantsin, A. Wolpert, M. Ceberio, G. Xiang, and V. Kreinovich, Detecting Outliers under Interval Uncertainty: A New Algorithm Based on Constraint Satisfaction, *Proceedings of the International Conference on Information Processing and Management*

of Uncertainty in Knowledge-Based Systems IPMU'06, Paris, France, Jul. 2-7, 2006, pp. 802–809.

[50] J. L. Davis and A. P. Annan, Ground Penetrating Radar for Soil and Rock Stratigraphy, *Geophysical Prospecting*, 1997, Vol. 37, pp. 531–551.

[51] P. Debroux, J. Boehm, F. Modave, V. Kreinovich, G. Xiang, J. Beck, K. Tupelly, R. Kandathi, L. Longpré, and K. Villaverde, Using 1-D Radar Observations to Detect a Space Explosion Core Among the Explosion Fragments: Sequential and Distributed Algorithms, *Proceedings of the 11th IEEE Digital Signal Processing Workshop*, Taos, New Mexico, Aug. 1-4, 2004, pp. 273–277.

[52] R. Demicco, J. Klir, Eds., *Fuzzy Logic in Geology*, Academic Press, 2003.

[53] L. Demkowicz, W. Rachowicz, P. Devloo, *A fully automatic hp-adaptivity*, The University of Texas at Austin, TICAM Report 01-28, 2001.

[54] A. P. Dempster, Upper and Lower Probabilities Induced by a Multi-Valued Mapping, *Ann. Mat. Stat.*, 1967, Vol. 38, pp. 325–339.

[55] O. Dessombz, F. Thouverez, J.-P. Laîné, and L. Jézéquel, Analysis of Mechanical Systems Using Interval Computations Applied to Finite Elements Methods, *J. Sound. Vib.*, 2001, Vol. 238, No. 5, pp. 949–968.

[56] D. I. Doser, K. D. Crain, M. R. Baker, V. Kreinovich, and M. C. Gerstenberger Estimating Uncertainties for Geophysical Tomography, *Reliable Computing*, 1998, Vol. 4, No. 3, pp. 241–268.

[57] R. E. Edwards, *Functional Analysis: Theory and Applications*, Dover Publ., N.Y., 1995.

[58] I. Elishakoff, and Y. Ren, The Bird's Eye View on Finite Element Method for Stochastic Structures, *Computer Methods in Applied Mechanics and Engineering*, 1999, Vol. 168, pp. 51–61.

[59] S. Ferson, *RAMAS Risk Calc: Risk Assessment with Uncertain Numbers*, Applied Biomathematics, Setauket, NY, 1999.

[60] S. Ferson, *RAMAS Risk Calc 4.0: Risk Assessment with Uncertain Numbers*, CRC Press, Boca Raton, Florida, 2002.

[61] S. Ferson, and L. R. Ginzburg, Different Methods are Needed to Propagate Ignorance and Variability, *Reliab. Engng. Syst. Saf.*, 1996, Vol. 54, pp. 133–144.

[62] S. Ferson, L. Ginzburg, V. Kreinovich, and M. Aviles, Exact Bounds on Sample Variance of Interval Data, *Extended Abstracts of the 2002 SIAM Workshop on Validated Computing*, Toronto, Canada, 2002, pp. 67–69.

[63] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, Exact Bounds on Finite Populations of Interval Data, *Reliable Computing*, 2005, Vol. 11, No. 3, pp. 207–233.

[64] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, Computing Variance for Interval Data is NP-Hard, *ACM SIGACT News*, 2002, Vol. 33, No. 2, pp. 108–118.

[65] S. Ferson, V. Kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz. *Constructing Probability Boxes and Dempster-Shafer structures*, Sandia National Laboratories, Technical Report SAND2002-4015, 2003.

[66] S. Ferson, D. Myers, and D. Berleant, *Distribution-free Risk Analysis: I. Range, Mean, and Variance*, Applied Biomathematics, Technical Report, 2001.

[67] S. Ferson, L. Ginzburg, V. Kreinovich, and J. Lopez, Absolute Bounds on the Mean of Sum, Product, etc.: A Probabilistic Extension of Interval Arithmetic, *Extended Abstracts of the 2002 SIAM Workshop on Validated Computing*, Toronto, Canada, May 23–25, 2002, pp. 70–72.

[68] E. Fisher, G. A. McMechan, and A. P. Annan, Acquisition and Processing of Wide-Aperture Ground-Penetrating Radar Data, *Geophysics*, 1992, Vol. 57, No. 3, pp. 495–504.

[69] W. A. Fuller, *Measurement Error Models*, J. Wiley & Sons, New York, 1987.

[70] J. Galambos, *The Asymptotic Theory of Extreme Order Statistics*, Wiley, New York, 1987.

[71] S. Ganzerli, and C. P. Pantelides, Load and Resistance Convex Models for Optimum Design, *Struct. Optim.*, 1999, Vol. 17, pp. 259–268.

[72] M. E. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of Np-Completeness*, Freeman, San Francisco, 1979.

[73] D. J. H. Garling, A Book Review, *American Mathematical Monthly*, 2005, Vol. 112, No. 6, pp. 575–579.

[74] L. Granvilliers, V. Kreinovich, and N. Müller, Novel Approaches to Numerical Software with Result Verification, In: R. Alt, A. Frommer, R. B. Kearfott, and W. Luther (eds.), *Numerical Software with Result Verification*, (International Dagstuhl Seminar, Dagstuhl Castle, Germany, Jan. 19–24, 2003), Springer Lectures Notes in Computer Science, 2004, Vol. 2991, pp. 274–305.

[75] E. D. Guy, J. J. Daniels, S. J. Radzevicius, and M. A. Vendl, Demonstration of Using Crossed Dipole Gpr Antenna For Site Characterization, *Geophysical Research Letters*, 1999, Vol. 26, No. 22. pp. 3421–3424.

[76] F. P. Haeni, Use of Ground-Penetrating Radar and Continuous Seismic-Reflection on Surface-Water Bodies in Environmental and Engineering Studies, *Journal of Environmental & Engineering Geophysics*, 1996, Vol. 1, No. 1, pp. 27–36.

[77] A. Haldar, and S. Mahadevan, *Reliability Assessment Using Stochastic Finite Element Analysis*. John Wiley & Sons, New York, 2000.

[78] E. Hansen, *Global Optimization Using Interval Analysis*. Marcel Dekker, Inc., New York, 1992.

[79] E. Hansen, Sharpness in Interval Computations, *Reliable Computing*, 1997, Vol. 3, pp. 7–29.

[80] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*, Cambridge University Press, 1988.

[81] S. Hein and A. Zakhor, Halftone to Continuous Tone Conversion of Error-Diffusion Coded Images, *IEEE Transactions on Image Processing*, Feb. 1995, Vol. 4, No. 2, pp. 208–216.

[82] W. Höhn, and H. D. Mittelmann, Some Remarks on the Discrete Maximum Principle for Finite Elements of Higher-Order, *Computing*, 1981, Vol. 27, pp. 145–154, .

[83] J. A. Hole: Nonlinear High-Resolution Three-Dimensional Seismic Travel Time Tomography. J. Geophysical Research, 1992, Vol. 97, pp. 6553–6562.

[84] J. Holt, M. Vendl, F. Baumgartner, S. Radziviscius, and J. J. Daniels, Brownfields Site-Characterization Using Geophysics: A Case History from East Chicago, *Transactions of the Symposium on the Application of Geophysics to Engineering and Environmental Problems: Soc. Eng. and Min. Expl. Geophys.*, Apr. 26–29, 1998, Chicago, IL.

[85] P. J. Huber, *Robust statistics*, Wiley, New York, 2004.

[86] Interval computations website
http://www.cs.utep.edu/interval-comp

[87] J. Jájá, *An Introduction to Parallel Algorithms*, Addison-Wesley, Reading, MA, 1992.

[88] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*, Springer-Verlag, London, 2001.

[89] E. T. Jaynes, Where do we Stand on Maximum Entropy?, in R. D. Levine and M. Tribus (eds.), *The Maximum Entropy Formalism*, M.I.T. Press, Cambridge, MA, 1979, pp. 15–118.

[90] E. T. Jaynes, *Probability Theory: The Logic of Science*, Cambridge University Press, Cambridge, Massachusetts, 2003.

[91] J. Karátson, and S. Korotov, Discrete Maximum Principles for Finite Element Solutions of Nonlinear Elliptic Problems with Mixed Boundary Conditions, *Numer. Math.* 2005, Vol. 99, pp. 669–698.

[92] G. E. Karniadakis, and S. J. Sherwin, *Spectral/hp Element Methods for CFD*, Oxford University Press, Oxford, 1999.

[93] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht, 1996.

[94] R. B. Kearfott and V. Kreinovich (eds.), *Applications of Interval Computations*, Kluwer, Dordrecht, 1996.

[95] R. B. Kearfott and V. Kreinovich, Beyond Convex? Global Optimization Is Feasible Only for Convex Objective Functions: A Theorem, *Journal of Global Optimization*, 2005, Vol. 33, No. 4, pp. 617–624.

[96] G. R. Keller, S. A. Starks, A. Velasco, M. Averill, R. Araiza, G. Xiang, and V. Kreinovich, Towards Combining Probabilistic, Interval, Fuzzy Uncertainty, and Constraints: ON the Example of Inverse Problem in Geophysics, *Proceedings of the Second International Conference on Fuzzy Sets and Soft Computing in Economics and Finance FSSCEF'2006*, St. Petersburg, Russia, Jun. 28–Jul. 1, 2006, pp. 47–54.

[97] D. G. Kendall, Foundations of a theory of random sets. In: E. Harding and D. Kendall (eds.), *Stochastic Geometry*. New York, pp. 322–376.

[98] T. D. Kite, N. Damera-Venkata, B. L. Evans, and A. C. Bovik, A Fast, High-Quality Inverse Halftoning Algorithm for Error Diffused Halftones, *IEEE Transactions on Image Processing*, Sep. 2000, Vol. 9, No. 9, pp. 1583–1592.

[99] S. Korotov, M. Křížek, and P. Neittaanmäki, Weakened Acute Type Condition for Tetrahedral Triangulations and the Discrete Maximum Principle, *Math. Comp.*, 2000, Vol. 70, pp. 107–119.

[100] O. Kosheleva, Symmetry-group Justification of Maximum Entropy Method and Generalized Maximum Entropy Methods in Image Processing, in: G. J. Erickson, J. T. Rychert, and C. R. Smith (eds.), *Maximum Entropy and Bayesian Methods*, Kluwer, Dordrecht, 1998, pp. 101–113.

[101] O. Kosheleva, On the Optimal Choice of Quality Metric In Image Compression: A Soft Computing Approach, *Soft Computing*, 2004, Vol. 8, No. 4, pp. 268–273.

[102] O. Kosheleva, S. Cabrera, B. Usevitch, and E. Vidal, Jr., Compressing 3D Measurement Data under Interval Uncertainty, *Proceedings of the Workshop on State-of-the-Art in Scientific Computing PARA'04*, Lyngby, Denmark, June 20–23, 2004, Vol. 1, pp. 79–85.

[103] S. Kotz and S. Nadarajah, *Extreme Value Distributions: Theory and Applications*, Imperial College Press, London, UK, 2000.

[104] U. Koyluoglu, S. Cakmak, N. Ahmet, and R. K. Soren, Interval Algebra to Deal with Pattern Loading and Structural Uncertainty, *J. Engrg. Mech.*, 1995, Vol. 121, No. 11, pp. 1149–1157.

[105] V. Kreinovich, Probabilities, Intervals, What Next? Optimization Problems Related

to Extension of Interval Computations to Situations with Partial Information about Probabilities, *Journal of Global Optimization*, 2004, Vol. 29, No. 3, pp. 265–280.

[106] V. Kreinovich and S. Ferson, Computing Best-Possible Bounds for the Distribution of a Sum of Several Variables is NP-hard, *International Journal of Approximate Reasoning*, 2006, Vol. 41, pp. 331–342.

[107] V. Kreinovich, S. Ferson, and L. Ginzburg, Exact Upper Bound on the Mean of the Product of Many Random Variables With Known Expectations, *Reliable Computing*, 2003, Vol. 9, No. 6, pp. 441–463.

[108] V. Kreinovich and L. Longpré, Computational Complexity and Feasibility Of Data Processing and Interval Computations, with Extension to Cases When We Have Partial Information about Probabilities, In: V. Brattka, M. Schroeder, K. Weihrauch, and N. Zhong, *Proceedings of the Conference on Computability and Complexity in Analysis CCA'2003*, Cincinnati, Ohio, USA, Aug. 28–30, 2003, pp. 19–54.

[109] V. Kreinovich, G. N. Solopchenko, S. Ferson, L. Ginzburg, and R. Aló, Probabilities, Intervals, What Next? Extension of Interval Computations to Situations with Partial Information about Probabilities, *Proceedings of the 10th IMEKO TC7 International Symposium on Advances of Measurement Science*, St. Petersburg, Russia, Jun. 30–Jul. 2, 2004, Vol. 1, pp. 137–142.

[110] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1997.

[111] V. Kreinovich and L. Longpré, Computational complexity and feasibility of data processing and interval computations, with extension to cases when we have partial information about probabilities, In: V. Brattka, M. Schroeder, K. Weihrauch, and N. Zhong, editors, *Proc. Conf. on Computability and Complexity in Analysis CCA'2003*, Cincinnati, Ohio, USA, Aug. 28–30, 2003, pp. 19–54.

[112] V. Kreinovich and L. Longpré, Fast Quantum Algorithms for Handling Probabilistic and Interval Uncertainty, *Mathematical Logic Quarterly*, 2004, Vol. 50, No. 4/5, pp. 507–518.

[113] V. Kreinovich, L. Longpré, S. Ferson, and L. Ginzburg, *Computing Higher Central Moments for Interval Data*, University of Texas at El Paso, Department of Computer Science, Technical Report UTEP-CS-03-14b, 2004, http://www.cs.utep.edu/vladik/2003/tr03-14b.pdf

[114] V. Kreinovich, L. Longpré, P. Patangay, S. Ferson, and L. Ginzburg, Outlier Detection Under Interval Uncertainty: Algorithmic Solvability and Computational Complexity, In: I. Lirkov, S. Margenov, J. Wasniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, Proceedings of the 4-th International Conference LSSC'2003, Sozopol, Bulgaria, Jun. 4–8, 2003, Springer Lecture Notes in Computer Science, 2004, Vol. 2907, pp. 238–245

[115] V. Kreinovich, L. Longpré, P. Patangay, S. Ferson, and L. Ginzburg, Outlier Detection Under Interval Uncertainty: Algorithmic Solvability and Computational Complexity, Reliable Computing, 2005, Vol. 11, No. 1, pp. 59–76.

[116] V. Kreinovich, H. T. Nguyen, and B. Wu, On-Line Algorithms for Computing Mean and Variance of Interval Data, and Their Use in Intelligent Systems, *Information Sciences*, 2007, Vol. 177, No. 16, pp. 3228–3238.

[117] V. Kreinovich, P. Patangay, L. Longpré, S. A. Starks, C. Campos, S. Ferson, and L. Ginzburg, Outlier Detection Under Interval and Fuzzy Uncertainty: Algorithmic Solvability and Computational Complexity, *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society NAFIPS'2003*, Chicago, Illinois, Jul. 24–26, 2003, pp. 401–406.

[118] V. Kreinovich, G. Xiang and S. Ferson, How the Concept of Information as Average Number of "Yes"-"No" Questions (Bits) Can be Extended to Intervals, P-Boxes, and

More General Uncertainty", *Proceedings of the 24th International Conference of the North American Fuzzy Information Processing Society NAFIPS'2005*, Ann Arbor, Michigan, Jun. 22-25, 2005, pp. 80–85.

[119] V. Kreinovich, G. Xiang, S. A. Starks, L. Longpré, M. Ceberio, R. Araiza, J. Beck, R. Kandathi, A. Nayak, R. Torres, and J. Hajagos, Towards Combining Probabilistic and Interval Uncertainty in Engineering Calculations: Algorithms for Computing Statistics under Interval Uncertainty, and Their Computational Complexity, *Reliable Computing*, 2006, Vol. 12, No. 6, pp. 471–501.

[120] V. Kreinovich, L. Longpré, S. A. Starks, G. Xiang, J. Beck, R. Kandathi, A. Nayak, S. Ferson, and J. Hajagos, Interval Versions of Statistical Techniques, with Applications to Environmental Analysis, Bioinformatics, and Privacy in Statistical Databases, to *Journal of Computational and Applied Mathematics*, 2007, Vol. 199, No. 2, pp. 418–423.

[121] V. Kreinovich, G. Xiang, and S. Ferson, Efficient Algorithms for Computing Mean and Variance Under Dempster-Shafer Uncertainty, *International Journal of Approximate Reasoning*, 2006, Vol. 42, pp. 212–227.

[122] M. Křížek, and L. Liu, On the maximum and comparison principles for a steady-state nonlinear heat conduction problem, *ZAMM Z. Angew. Math. Mech.*, 2003, Vol. 83, pp. 559–563.

[123] V. P. Kuznetsov, *Interval Statistical Models*, Radio i Svyaz, Moscow, 1991 (in Russian).

[124] A. T. Langewisch and F. F. Choobineh, Mean and variance bounds and propagation for ill-specified random variables, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 2004, Vol. 34, No. 4, pp. 494–506.

[125] W. K. Liu, T. Belytschko, and A. Mani, Probabilistic Finite Elements for Nonlinear Structural Dynamics. *Computer Methods in Applied Mechanics and Engineering*, 1986, Vol. 56, pp. 61–81.

[126] W. A. Lodwick, and K. D. Jamison, Special issue: interface between fuzzy set theory and interval analysis. *Fuzzy Sets and Systems*, 2002, Vol. 135, No. 1–3.

[127] W. A. Lodwick and K. D. Jamison, Estimating and Validating the Cumulative Distribution of a Function of Random Variables: Toward the Development of Distribution Arithmetic, *Reliable Computing*, 2003, Vol. 9, No. 2, pp. 127–141.

[128] M. Maceira, S. R. Taylor, C. J. Ammon, X. Yang, A. A. Velasco, High-resolution Rayleigh wave slowness tomography of Central Asia. Journal of Geophysical Research, 2005, Vol. 110, paper B06304.

[129] M. Martinez, L. Longpré, V. Kreinovich, S. A. Starks, and H. T. Nguyen, Fast Quantum Algorithms for Handling Probabilistic, Interval, and Fuzzy Uncertainty, *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society NAFIPS'2003*, Chicago, Illinois, Jul. 24–26, 2003, pp. 395–400.

[130] S. McWilliam, Anti-optimisation of uncertain structures using interval analysis. *Comput. Struct.*, 2000, Vol. 79, pp. 421–430.

[131] R. E. Melchers, *Structural Reliability Analysis and Prediction*, 2nd Edition, John Wiley & Sons, West Sussex, England, 1999.

[132] M. Mese and P. P. Vaidyanathan, Optimized halftoning using dot diffusion and methods for inverse halftoning, *IEEE Transactions on Image Processing*, Apr. 2000, Vol. 9, No. 4, pp. 691–709.

[133] F. Modave, M. Ceberio, X. Wang, G. Xiang, O. Garay, R. Ramirez and R. Tejeda, Comparison of Computer Attacks: An Application of Interval-based Fuzzy Integration, *Proceedings of the 24th International Conference of the North American Fuzzy*

Information Processing Society NAFIPS'2005, Ann Arbor, Michigan, Jun. 22-25, 2005, pp. 676–681.

[134] N. Moës, J. Dolbow, and T. Belytschko. A Finite Element Method for Crack Growth Without Remeshing, *Int. J. Numer. Methods Engrg.*, 1999, Vol. 46, pp. 131–150.

[135] B. Möller, W. Graf, and M. Beer, Fuzzy structural analysis using level-optimization. *Comput. Mech.*, 2000, Vol. 26, No. 6, pp. 547–565.

[136] R. E. Moore, *Automatic error analysis in digital computation*, Technical Report Space Div. Report LMSD84821, Lockheed Missiles and Space Co., 1959.

[137] R. E. Moore, *Interval Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1966.

[138] R. E. Moore, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

[139] R. E. Moore and W. A. Lodwick, Interval Analysis and Fuzzy Set Theory, *Fuzzy Sets and Systems*, 2003, Vol. 135, No. 1, pp. 5–9.

[140] D. Morgenstein and V. Kreinovich, Which algorithms are feasible and which are not depends on the geometry of space-time, *Geombinatorics*, 1995, Vol.4, No.3, pp. 80–97.

[141] R. Muhanna, V. Kreinovich, P. Solin, J. Chessa, R. Araiza, and G. Xiang, Interval Finite Element Methods: New Directions, *Proceedings of the Second International Workshop on Reliable Engineering Computing*, Savannah, Georgia, Feb. 22–24, 2006, pp. 229–243.

[142] R. L. Muhanna, and R. L. Mullen, Development of interval based methods for fuzziness in continuum mechanics. In: *Proc. ISUMA-NAFIPS'95*, 1995, pp. 23–45.

[143] R. L. Muhanna, and R. L. Mullen, Formulation of fuzzy finite element methods for mechanics problems. *Compu.-Aided Civ. Infrastruct. Engrg.*, 1999, Vol. 14, pp. 107–117.

[144] R. L. Muhanna, and R. L. Mullen, Uncertainty in mechanics problems-interval-based approach. *J. Engrg. Mech.*, 2001, Vol. 127, No.6, pp. 557–566.

[145] R. Muhanna and R. Mullen (eds.), *Proceedings of the NSF Workshop on Reliable Engineering Computing*, Savannah, Georgia, Sep. 15–17, 2004.

[146] R. L. Muhanna, R. L. Mullen, and H. Zhang, Penalty-Based Solution for the Interval Finite-Element Methods, *ASCE, Engineering Mechanics*, 2005, Vol. 131, No. 10, pp. 1102–1111.

[147] R. L. Mullen, and R. L. Muhanna, Structural analysis with fuzzy-based load uncertainty. In: *Proc. 7th ASCE EMD/STD Joint Spec. Conf. on Probabilistic Mech. and Struct. Reliability.* Mass., 1996, pp. 310–313.

[148] R. L. Mullen, and R. L. Muhanna, Bounds of structural response for all possible loadings. *J. Struct. Engrg., ASCE*, 1999, Vol. 125, No.1, pp. 98–106.

[149] A. Neumaier, *Interval Methods for Systems of Equations.* Cambridge University Press, Cambridge, 1990.

[150] A. Neumaier and A. Pownuk, Linear systems with large uncertainties, with applications to truss structures, *Reliable Computing*, 2007, Vol. 13, No. 2, pp. 149–172.

[151] H. T. Nguyen and V. Kreinovich, Nested Intervals and Sets: Concepts, Relations to Fuzzy Sets, and Applications, In: R. B. Kearfott and V. Kreinovich, editors, *Applications of Interval Computations*, Kluwer, Dordrecht, 1996, pp. 245–290.

[152] H. T. Nguyen and V. Kreinovich, *Applications of continuous mathematics to computer science*, Kluwer, Dordrecht, 1997.

[153] H. T. Nguyen and E. A. Walker, *First Course in Fuzzy Logic*, CRC Press, Boca Raton, Florida, 1999.

[154] H. T. Nguyen, T. Wang, and V. Kreinovich, Towards Foundations of Processing Imprecise Data: From Traditional Statistical Techniques of Processing Crisp Data to Statistical Processing of Fuzzy Data, In: Y. Liu, G. Chen, M. Ying, and K.-Y. Cai, editors, *Proceedings of the International Conference on Fuzzy Information Processing: Theories and Applications FIP'2003*, Beijing, China, Mar. 1–4, 2003, Vol. 2, pp. 895–900.

[155] H. T. Nguyen, B. Wu, and V. Kreinovich, Shadows of Fuzzy Sets – A Natural Approach Towards Describing 2-D and Multi-D Fuzzy Uncertainty in Linguistic Terms, *Proc. 9th IEEE Int'l Conference on Fuzzy Systems FUZZ-IEEE'2000*, San Antonio, Texas, May 7–10, 2000, Vol. 1, pp. 340–345.

[156] H. T. Nguyen, V. Kreinovich, and G. Xiang, Foundations of Statistical Processing of Set-valued Data: Towards Efficient Algorithms, *Proceedings of the Fifth International Conference on Intelligent Technologies InTech'04*, Houston, Texas, Dec. 2–4, 2004.

[157] P. Nivlet, F. Fournier, and J. Royer, A new methodology to account for uncertainties in 4-D seismic interpretation, *Proc. 71st Annual Int'l Meeting of Soc. of Exploratory Geophysics SEG'2001*, San Antonio, TX, Sep. 9–14, 2001, pp. 1644–1647.

[158] P. Nivlet, F. Fournier, and J. Royer, Propagating interval uncertainties in supervised pattern recognition for reservoir characterization, *Proc. 2001 Society of Petroleum Engineers Annual Conf. SPE'2001*, New Orleans, LA, Sep. 30–Oct. 3, 2001, paper SPE-71327.

[159] P. V. Novitskii and I. A. Zograph, *Estimating the Measurement Errors*, Energoatomizdat, Leningrad, 1991 (in Russian).

[160] G. R. Olhoeft, Electrical, magnetic, and geometric properties that determine ground penetrating performance, In: *Proc. Seventh International Conference on Ground Penetrating Radar*, the University of Kansas, Lawrence, KS, May 27–30, 1998.

[161] M. Orshansky, Increasing circuit performance through statistical design techniques, in: D. Chinnery and K. Keutzer (eds.), *Closing the Gap Between ASICs and Custom*, Kluwer, 2002.

[162] M. Orshansky and A. Bandyopadhyay, Fast statistical timing analysis handling arbitrary delay correlations, *Proc. of the Design Automation Conference DAC'2004*, San Diego, California, Jun. 7–11, 2004, pp. 337–342.

[163] M. Orshansky and K. Keutzer, A general probabilistic framework for worst case timing analysis, *Proceedings of the Design Automation Conference DAC'2002*, Jun. 2002, pp. 556–561.

[164] M. Orshansky and K. Keutzer, From Blind Certainty to Informed Uncertainty, *Proceedings of the TAU Workshop*, Dec. 2002.

[165] M. Orshansky, W. Wang, M. Ceberio, and G. Xiang, Interval-based Robust Statistical Techniques for Non-negative Convex Functions, with Application to Timing Analysis of Computer Chips, *Proceedings of the ACM Symposium on Applied Computing SAC'06*, Dijon, France, Apr. 23-27, 2006, pp. 1629–1633.

[166] M. Orshansky, W. Wang, G. Xiang, and V. Kreinovich, Interval-based Robust Statistical Techniques for Non-negative Convex Functions with Application to Timing Analysis of Computer Chips, *Proceedings of the Second International Workshop on Reliable Engineering Computing*, Savannah, Georgia, Feb. 22–24, 2006, pp. 197–212.

[167] R. Osegueda, V. Kreinovich, L. Potluri, R. Aló, Non-Destructive Testing of Aerospace Structures: Granularity and Data Mining Approach, *Proc. FUZZ-IEEE'2002*, Honolulu, HI, May 12–17, 2002, Vol. 1, pp. 685–689

[168] C. P. Pantelides, and S. Ganzerli, Comparison of fuzzy set and convex model theories in structural design. *Mech. Systems Signal Process.*, 2001, Vol. 15, No.3, pp. 499–511.

[169] C. H. Papadimitriou, *Computational Complexity*, Addison Wesley, San Diego, 1994.

[170] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, Inc., Mineola, New York, 1998.

[171] R. L. Parker, *Geophysical Inverse Theory*, Princeton University Press, Princeton, New Jersey, 1994.

[172] M. Paszynski, J. Kurtz, L. Demkowicz, *Parallel, Fully Automatic hp-Adaptive 2D Finite Element Package*, The University of Texas at Austin, TICAM Report 04-07, 2004.

[173] E. D. Popova, M. Datcheva, R. Iankov, and T. Schanz. Mechanical models with interval parameters. In: Gürlebeck, G., L. Hempel, C. Könke, editors, *IKM2003: Digital Proceedings of 16th International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering*, Weimar, Germany, 2003.

[174] *Proceedings of the 10th International Conference on Ground Penetrating Radar GPR'2004*, Delft, The Netherlands, Jun. 21–24, 2004.

[175] S. Rabinovich, *Measurement Errors: Theory and Practice*, American Institute of Physics, New York, 1993.

[176] W. Rachowicz, D. Pardo, L. Demkowicz, *Fully Automatic hp-Adaptivity in Three Dimensions*, The University of Texas at Austin, ICES Report 04-22, 2004.

[177] S. Rajasekaran, P. Pardalos, J. Reif, and J. Rolim (eds.), *Handbook on Randomized Computing*, Kluwer, 2001.

[178] S. S. Rao and L. Berke, Analysis of uncertain structural systems using interval analysis. *AIAA J.*, 1997, Vol. 35, No. 4, pp. 727–735.

[179] S. S Rao and L. Chen, Numerical solution of fuzzy linear equations in engineering analysis. *Int. J. Numer. Meth. Engng.*, 1998, Vol. 43, pp. 391–408.

[180] H. Regan, S. Ferson, and D. Berleant, Equivalence of five methods for bounding uncertainty, *Journal of Approximate Reasoning*, 2004, Vol. 36, No. 1, pp. 1–30.

[181] N. C. Rowe, Absolute bounds on the mean and standard deviation of transformed data for constant-sign-derivative transformations, *SIAM Journal of Scientific Statistical Computing*, 1988, Vol.9, pp. 1098–1113.

[182] S. M. Rump, Solving algebraic problems with high accuracy. In: Kulisch, U., and W. Miranker (eds.): *A New Approach to Scientific Computation*. Academic Press, New York, 1983.

[183] G. Schuëller, Computational Stochastic Mechanics – Recent Advances, *Computers and Structures*, 2001, Vol. 79, pp. 2225–2234.

[184] K. Sentz, and S. Ferson, *Combination of Evidence in Dempster-Shafer Theory*, Sandia National Laboratories, Technical Report SAND2002-0835, 2002.

[185] J. A. Sethian, *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[186] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, New Jersey, 1976.

[187] I. Shmulevich and W. Zhang, Binary analysis and optimization-based normalization of gene expression data, *Bioinformatics*, 2002, Vol. 18, No. 4, pp. 555–565.

[188] P. Šolín,*Partial Differential Equations and the Finite Element Methods*, J. Wiley & Sons, Hoboken, New Jersey, 2005.

[189] P. Šolín and L. Demkowicz, Goal-Oriented *hp*-Adaptivity for Elliptic Problems, *Computer Methods in Applied Mechanics and Engineering*, 2004, Vol. 193, pp. 449–468.

[190] P. Šolín, K. Segeth, and I. Doležel, *Higher-Order Finite Element Methods*, Chapman & Hall/CRC Press, Boca Raton, 2003.

[191] P. Šolín and T. Vejchodský, *On the Discrete Maximum Principle for the hp-FEM*, University of Texas at El Paso, Department of Mathematical Science, Technical Report, February 2005, http://www.math.utep.edu/Faculty/solin/new_papers/dmp.pdf; see also http://www.math.utep.edu/Faculty/solin/new_papers/dmp-coll.pdf

[192] P. Šolín, T. Vejchodský, and R. Araiza, *Discrete Conservation of Nonnegativity or Elliptic Problems Solved by the hp-FEM*, University of Texas at El Paso, Department of Computer Science, Technical Report UTEP-CS-05-29, Aug. 2005, http://www.cs.utep.edu/vladik/2005/tr05-29.pdf

[193] S. Srikrishnan, R. Araiza, V. Kreinovich, S. A. Starks, and H. Xie, Automatic Referencing of Satellite and Radar Images, *Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference*, Tucson, Arizona, Oct. 7–10, 2001, pp. 2176–2181.

[194] S. A. Starks, V. Kreinovich, L. Longpré, M. Ceberio, G. Xiang, R. Araiza, J. Beck, R. Kandathi, A. Nayak, and R. Torres, Towards combining probabilistic and interval uncertainty in engineering calculations, *Proceedings of the Workshop on Reliable Engineering Computing*, Savannah, Georgia, Sep. 15–17, 2004, pp. 193–213.

[195] F.Stazi, E. Budyn, J. Chessa, and T. Belytschko, An Extended Finite Element Method With Higher-Order Elements for Crack Problems With Curvature, *Computational Mechanics*, 2003, Vol. 31, No.1–2, pp. 38–48.

[196] N. Sukumar, D. L. Chopp, N. Moës, and T. Belytschko, Modeling Holes and Inclu-

sions by Level Sets in the Extended Finite Element Method, *Computer Methods in Applied Mechanics and Engineering*, 2000, Vol. 190, No.46–47, pp. 6183–6200.

[197] A. Tarantola, *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*, Elsevier, Amsterdam, 1987.

[198] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*, Kluwer, Boston, Dordrecht, London, 2002.

[199] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*, W. H. Whinston & Sons, Washington, D.C., 1977.

[200] R. Trejo and V. Kreinovich, Error Estimations for Indirect Measurements: Randomized vs. Deterministic Algorithms For 'Black-Box' Programs, In: S. Rajasekaran, P. Pardalos, J. Reif, and J. Rolim (eds.), *Handbook on Randomized Computing*, Kluwer, 2001, pp. 673-729.

[201] S. A. Vavasis, *Nonlinear Optimization: Complexity Issues*, Oxford University Press, New York, 1991.

[202] H. M. Wadsworth, Jr., editor, *Handbook of statistical methods for engineers and scientists*, McGraw-Hill Publishing Co., N.Y., 1990.

[203] P. Walley, *Statistical Reasoning with Imprecise Probabilities*, Chapman & Hall, N.Y., 1991.

[204] G. W. Walster, Philosophy and practicalities of interval arithmetic, In: *Reliability in Computing*, Academic Press, N.Y., 1988, pp. 309–323.

[205] G. W. Walster and V. Kreinovich, For unknown-but-bounded errors, interval estimates are often better than averaging, *ACM SIGNUM Newsletter*, 1996, Vol. 31, No. 2, pp. 6–19.

[206] R. Williamson and T. Downs, Probabilistic arithmetic I: numerical methods for calculating convolutions and dependency bounds, *International Journal of Approximate Reasoning*, 1990, Vol. 4, pp. 89–158.

[207] P. W. Wong, Image quantization, halftoning, and printing, In: A. Bovik (ed.), *Handbook of Image and Video Processing*, Academic Press, New York, 2000, pp. 657–667.

[208] B. Wu, H. T. Nguyen, and V. Kreinovich, Real-Time Algorithms for Statistical Analysis of Interval Data, *Proceedings of the International Conference on Information Technology InTech'03*, Chiang Mai, Thailand, Dec. 17–19, 2003, pp. 483–490.

[209] G. Xiang, S. A. Starks, V. Kreinovich, and L. Longpré, New Algorithms for Statistical Analysis of Interval Data, *Proceedings of the Workshop on State-of-the-Art in Scientific Computing PARA'04*, Lyngby, Denmark, Jun. 20–23, 2004, Vol. 1, pp. 123–129.

[210] G. Xiang, Scott A. Starks, V. Kreinovich, and L. Longpré, New Algorithms for Statistical Analysis of Interval Data, In: J. Dongarra, K. Madsen, and J. Wasniewski editor, *PARA'04 Workshop on State-of-the-Art in Scientific Computing*, Springer Lecture Notes in Computer Science, 2005, Vol. 3732, pp. 189–196.

[211] G. Xiang, O. Kosheleva, and G. J. Klir, Estimating Information Amount Under Interval Uncertainty: Algorithmic Solvability and Computational Complexity, *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'06*, Paris, France, Jul. 2–7, 2006, pp. 840–847.

[212] G. Xiang, Fast algorithm for computing the upper endpoint of sample variance for interval data: case of sufficiently accurate measurements, *Reliable Computing*, 2006, Vol. 12, No. 1, pp. 59–64.

[213] L. A. Zadeh, Fuzzy Sets as a Basis for a Theory of Possibility. *Fuzzy Sets and Systems*, 1978, Vol. 1, pp. 3–28.

[214] C. A. Zelt, P. J. Barton, Three-dimensional seismic refraction tomography: A comparison of two methods applied to data from the Faeroe Basin. J. Geophysical Research 1998, Vol. 103, pp. 7187–7210.

[215] W. Zhang, I. Shmulevich, and J. Astola, *Microarray Quality Control*, Wiley, Hoboken, New Jersey, 2004.

Curriculum Vitae

Gang Xiang was born on February 16, 1979 in Hefei, Anhui Province, China. He is the only child of Qilian Xiang and Jiacai Zhong. After graduating from Hefei High School No. 1 in 1997, he entered Zhejiang University at Hangzhou, Zhejiang Province, China and received a Bachelor of Engineering degree in Computer Science and Technology in June 2001. Then he worked at China Telecom at Shenzhen, Guangdong Province, China from July 2001 until August 2003. In 2003, he enrolled at the University of Texas at El Paso to pursue a doctorate degree in Computer Science.

His main research interest is to study challenging theoretical computer science problems motivated by application needs. His main expertise and experience have been in issues related to uncertainty, especially interval uncertainty and practically important combinations of interval and probabilistic uncertainty. He has developed and applied new algorithms and uncertainty techniques to computer security (privacy-preserving data analysis), computer engineering (chip design), geosciences, defense-related signal processing and finite element methods. He also has some experience with decision making and web mining.

He published more than 20 papers, most of them in international journals and proceedings of international conferences.

He is an Associate Member of Sigma Xi, the Scientific Research Society.

Permanent address: 410 South Tongcheng Road 3-503, Baohe District
Hefei, Anhui Province, P.R.China, 230000