Computing Mean and Variance Under Dempster-Shafer Uncertainty: Towards Faster Algorithms

Vladik Kreinovich[1], Gang Xiang[1], and Scott Ferson[2]

[1]Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA, vladik@utep.edu
[2]Applied Biomathematics, 100 North Country Road
Setauket, NY 11733, USA, scott@ramas.com

Abstract

In many real-life situations, we only have partial information about the actual probability distribution. For example, under Dempster-Shafer uncertainty, we only know the masses $m_1, \ldots, m_n$ assigned to different sets $S_1, \ldots, S_n$, but we do not know the distribution within each set S_i. Because of this uncertainty, there are many possible probability distributions consistent with our knowledge; different distributions have, in general, different values of standard statistical characteristics such as mean and variance. It is therefore desirable, given a Dempster-Shafer knowledge base, to compute the *ranges* $[\underline{E}, \overline{E}]$ and $[\underline{V}, \overline{V}]$ of possible values of mean E and of variance V.

In their recent paper, A. T. Langewisch and F. F. Choobineh show how to compute these ranges in polynomial time. In particular, they reduce the problem of computing $\overline{V}$ to the problem of minimizing a convex quadratic function, a problem which can be solved in time $O(n^2 \cdot \log(n))$. We show that the corresponding quadratic optimization problem can be actually solved faster, in time $O(n \cdot \log(n))$; thus, we can compute the bounds $\underline{V}$ and $\overline{V}$ in time $O(n \cdot \log(n))$.

1 Formulation of the Problem

Computing mean and variance under Dempster-Shafer uncertainty: an important practical problem. In many real-life situations, we only have partial information about the actual probability distribution. In many practical situations, this uncertainty is naturally described by a Dempster-Shafer (DS) approach (see, e.g., [9]). In the 1-D case, instead of the exact probability distribution, we have a finite collection of intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$, ..., $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$, and we have non-negative "masses" (probabilities) $m_1, \ldots, m_n$ assigned to these intervals in such a way that $m_1 + \ldots + m_n = 1$. This means that:

- with probability m_1, we select the interval $\mathbf{x}_1$,
- with probability m_2, we select the interval $\mathbf{x}_2$,
- ...
- with probability m_n, we select the interval $\mathbf{x}_n$;

then, within the selected interval $\mathbf{x}_i$, we select a value x according to some probability distribution $\rho_i(x)$ located on this interval. As a result, the overall probability distribution takes the form

$$\rho(x) = m_1 \cdot \rho_1(x) + \ldots + m_n \cdot \rho_n(x), \tag{1}$$

with $\rho_i(x)$ located on the interval $\mathbf{x}_i$.

For different distributions ρ_i, we get, in general, different resulting distributions $\rho(x)$, and thus, different values of statistical characteristics such as mean E and variance V. A natural question is: what are the ranges $[\underline{E}, \overline{E}]$ and $[\underline{V}, \overline{V}]$ of possible values of mean E and of variance V?

Known efficient algorithms for computing mean and variance under Dempster-Shafer uncertainty. Efficient algorithms for computing such ranges were described by A. T. Langewisch and F. F. Choobineh in their recent paper [6]. Namely, they showed that the bounds for the mean have the form

$$\underline{E} = \sum_{i=1}^{n} m_i \cdot \underline{x}_i, \quad \overline{E} = \sum_{i=1}^{n} m_i \cdot \overline{x}_i. \tag{2}$$

The upper bound $\overline{V}$ for the variance is equal to the maximum of the following concave optimization problem:

$$\overline{V} = \max \left(\sum_{i=1}^{n} \left(\underline{m}_i \cdot (\underline{x}_i)^2 + \overline{m}_i \cdot (\overline{x}_i)^2 \right) - \left(\sum_{i=1}^{n} (\underline{m}_i \cdot \underline{x}_i + \overline{m}_i \cdot \overline{x}_i) \right)^2 \right) \tag{3}$$

under the constraints

$$\underline{m}_i + \overline{m}_i = m_i, \quad \underline{m}_i \geq 0, \quad \overline{m}_i \geq 0, \text{ for all } i, \tag{4}$$

and the lower bound $\underline{V}$ is equal to the minimum of the following convex quadratic function:

$$\underline{V} = \min \left(\sum_{i=1}^{n} m_i \cdot x_i^2 - \left(\sum_{i=1}^{n} m_i \cdot x_i \right)^2 \right) \tag{5}$$

under the constraints

$$\underline{x}_i \le x_i \le \overline{x}_i \text{ for all } i. \tag{6}$$

(These cases correspond to Case IIb of [6]).

Formulas (2) enable us to compute the bounds $\underline{E}$ and $\overline{E}$ in linear time $O(n)$. For computing the bounds on the variance, the corresponding quadratic optimization problems can be solved in polynomial time – namely, in time $O(n^2 \cdot \log(n))$; see, e.g., [7]. Thus, A. T. Langewisch and F. F. Choobineh come up with efficient (polynomial time) algorithms for computing $\underline{V}$ and $\overline{V}$.

Our contribution. In this paper, we show that the quadratic optimization problems (3) and (5) can be solved even faster: in time $O(n\log(n))$. Thus, we have faster algorithms for computing the range $[\underline{V}, \overline{V}]$ of the variance V.

Structure of the paper. The algorithms are described in Section 2. In Section 3, we present a numerical example illustrating these algorithms. Section 4 contains the proof that the proposed algorithms indeed solve the desired quadratic optimization problems and that they indeed produce the solutions to these problems in time $O(n \cdot \log(n))$.

Comment: how good are the new algorithms? Since even simple sorting requires at least $O(n \cdot \log(n))$ steps (see, e.g., [1]), algorithms like this, that compute a bound of a statistical interval characteristic in $O(n \cdot \log(n))$ steps, can be considered a "golden standard" for such algorithms.

Comment. It is worth mentioning that a similar problem of *interval uncertainty*, where we know n intervals $\mathbf{x}_i$ and we want to find the range $[\underline{V}, \overline{V}]$ of possible values of sample variance

$$V = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - E)^2, \text{ where } E = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i, \tag{7}$$

is NP-hard; see, e.g., [2, 3, 4]. (Crudely speaking, NP-hard means that in general, we cannot compute the exact range $[\underline{V}, \overline{V}]$ faster than in exponential time $\approx 2^n$.) A simple numerical example explaining why interval and Dempster-Shafer bounds on the variance are different is given in the Appendix.

2 Algorithms

Algorithm for computing $\underline{V}$. The algorithm $\underline{\mathcal{V}}$ for computing $\underline{V}$ is as follows:

- First, we sort all $2n$ values $\underline{x}_i$, $\overline{x}_i$ into a sequence $x_{(1)} < x_{(2)} < \ldots < x_{(q)}$ for some $q \le 2n$. We will take $x_{(q+1)} \stackrel{\text{def}}{=} +\infty$.

- Second, we use bisection to find the value k $(1 \le k \le q)$ for which the following two inequalities hold:

$$\sum_{j:\overline{x}_j \le x_{(k)}} m_j \cdot (x_{(k)} - \overline{x}_j) \le \sum_{i:\underline{x}_i \ge x_{(k+1)}} m_i \cdot (\underline{x}_i - x_{(k)}); \tag{8}$$

$$\sum_{j:\overline{x}_j \le x_{(k)}} m_j \cdot (x_{(k+1)} - \overline{x}_j) > \sum_{i:\underline{x}_i \ge x_{(k+1)}} m_i \cdot (\underline{x}_i - x_{(k+1)}). \tag{9}$$

  At each iteration of this bisection, we have an interval $[k^-, k^+]$ that is guaranteed to contain k. In the beginning, $k^- = 1$ and $k^+ = q$. At each iteration, we compute the midpoint $k_{\rm mid} = \lfloor (k^- + k^+)/2 \rfloor$, and check both inequalities (8) and (9) for $k = k_{\rm mid}$. Then:

  - If both inequalities (8) and (9) hold for his k, this means that we have found the desired k.
  - If (8) holds but (9) does not hold, this means that the desired value k is larger than $k_{\rm mid}$, so we keep k^+ and replace k^- with $k_{\rm mid} + 1$.
  - If (9) holds but (8) does not hold, this means that the desired value k is smaller than $k_{\rm mid}$, so we keep k^- and replace k^+ with $k_{\rm mid} - 1$.

- Once k is found, we compute

$$S_k \stackrel{\rm def}{=} \sum_{j:\overline{x}_j \le x_{(k)}} m_j \cdot \overline{x}_j + \sum_{i:\underline{x}_i \ge x_{(k+1)}} m_i \cdot \underline{x}_i, \tag{10}$$

  and

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

  If $\Sigma_k = 0$, we take $\underline{V} = 0$; otherwise, we compute $r_k = S_k / \Sigma_k$, and then

$$\underline{V} = \sum_{j:\overline{x}_j \le x_{(k)}} m_j \cdot (\overline{x}_j - r_k)^2 + \sum_{i:\underline{x}_i \ge x_{(k+1)}} m_i \cdot (\underline{x}_i - r_k)^2.$$

Comment. In principle, it is possible that for all the values i, we have $\underline{x}_i < x_{(k+1)}$ and $x_{(k)} < \overline{x}_i$. In this case, Σ_k is the sum of an empty number of terms, i.e., by a usual definition of such a sum, $\Sigma_k = 0$. In this case, $\underline{V}$ is also the sum of an empty set of terms, i.e., 0.

Algorithm for computing $\overline{V}$. The algorithm $\overline{\mathcal{V}}$ for computing $\overline{V}$ is as follows:

- First, we sort all n midpoints $\widetilde{x}_i = \dfrac{1}{2} \cdot (\underline{x}_i + \overline{x}_i)$ into a non-decreasing sequence. After this sorting, we can assume that the intervals $\mathbf{x}_i$ are sorted in such a way that $\widetilde{x}_1 \le \widetilde{x}_2 \le \ldots \le \widetilde{x}_n$. We take $\widetilde{x}_{n+1} = +\infty$.

  We say that k is *proper* if $\widetilde{x}_k > \widetilde{x}_{k-1}$ or $k = 1$.

For each k, we denote by $l(k)$ the largest value l for which $\widetilde{x}_l = \widetilde{x}_k$, and by $s(k)$, the smallest value s for which $\widetilde{x}_s = \widetilde{x}_k$. (Hence, the value $s(k)$ is always proper.)

- Second, we use bisection to find the value k ($1 \le k \le n$) for which the following two inequalities hold (if such a value exists):

$$\sum_{j=k}^{n} m_j \cdot (\overline{x}_j - \widetilde{x}_k) < \sum_{i=1}^{k-1} m_i \cdot (\widetilde{x}_k - \underline{x}_i); \tag{11}$$

$$\sum_{j=k}^{n} m_j \cdot (\overline{x}_j - \widetilde{x}_{k-1}) \ge \sum_{i=1}^{k-1} m_i \cdot (\widetilde{x}_{k-1} - \underline{x}_i). \tag{12}$$

  At each iteration of this bisection, we have an interval $[k^-, k^+]$ that is guaranteed to contain k. In the beginning, $k^- = 1$ and $k^+ = n+1$. At each iteration, we compute the midpoint $k_{\mathrm{mid}} = \lfloor (k^- + k^+)/2 \rfloor$, and check both inequalities (11) and (12) for $k = k_{\mathrm{mid}}$. Then:

  - If both inequalities (11) and (12) hold for this k_{mid}, this means that we have found the desired k.
  - If, for k_{mid}, (11) holds but (12) does not hold, this means that the desired value k is smaller than k_{mid}, so we keep k^- and replace k^+ with $k_{\mathrm{mid}} - 1$.
  - If, for k_{mid}, (12) holds but (11) does not hold, this means that the desired value k is larger than k_{mid}, so we keep k^+ and replace k^- with $k_{\mathrm{mid}} + 1$.

  Once k is found, we compute

$$E \stackrel{\text{def}}{=} \sum_{i=1}^{k-1} m_i \cdot \underline{x}_i + \sum_{j=k}^{n} m_j \cdot \overline{x}_j,$$

  and then

$$V = \sum_{i=1}^{k-1} m_i \cdot (\underline{x}_i - E)^2 + \sum_{j=k}^{n} m_j \cdot (\overline{x}_j - E)^2.$$

- Third, we use bisection to find the proper value k ($1 \le k \le n$) for which the following two inequalities hold:

$$\sum_{j=l(k)+1}^{n} m_j \cdot (\overline{x}_j - \widetilde{x}_k) \le \sum_{i=1}^{l(k)} m_i \cdot (\widetilde{x}_k - \underline{x}_i); \tag{13}$$

$$\sum_{j=k}^{n} m_j \cdot (\overline{x}_j - \widetilde{x}_k) \ge \sum_{i=1}^{k-1} m_i \cdot (\widetilde{x}_k - \underline{x}_i). \tag{14}$$

At each iteration of this bisection, we have an interval $[k^-, k^+]$ that is guaranteed to contain k. In the beginning, $k^- = 1$ and $k^+ = n$. At each iteration, we compute the proper index $k_{\rm mid} = s(\lfloor (k^- + k^+)/2 \rfloor)$ corresponding to the midpoint, and check both inequalities (13) and (14) for the proper value $k = k_{\rm mid}$. Then:

- If both inequalities (13) and (14) hold for this $k_{\rm mid}$, this means that we have found the desired k.
- If, for $k_{\rm mid}$, (13) holds but (14) does not hold, this means that the desired value k is smaller than $k_{\rm mid}$, so we keep k^- and replace k^+ with $k_{\rm mid} - 1$.
- If, for $k_{\rm mid}$, (14) holds but (13) does not hold, this means that the desired value k is larger than $k_{\rm mid}$, so we keep k^+ and replace k^- with $l(k_{\rm mid}) + 1$.

Once k is found, we compute

$$V = \sum_{i=1}^{k-1} m_i \cdot (\underline{x}_i - \widetilde{x}_k)^2 + \sum_{j=k}^{n} m_j \cdot (\overline{x}_j - \widetilde{x}_k)^2.$$

- Finally, as $\overline{V}$, we take the largest of the two values of V obtained on the second and on the third stages.

3 Numerical Example

Case study. To clarify how our algorithms work, let us illustrate them on an example in which we have three intervals:

- the interval $[\underline{x}_1, \overline{x}_1] = [0, 5]$ with the mass $m_1 = \frac{1}{2}$,
- the interval $[\underline{x}_2, \overline{x}_2] = [1, 2]$ with mass $m_2 = \frac{1}{4}$, and
- the interval $[\underline{x}_3, \overline{x}_3] = [3, 4]$ with mass $m_1 = \frac{1}{4}$.

Computing $\underline{V}$. In accordance with the above algorithms, first, we sort all $2n = 6$ endpoints $\underline{x}_i$, $\overline{x}_i$ into a sequence:

$$0 = x_{(1)} < 1 = x_{(2)} < 2 = x_{(3)} < 3 = x_{(4)} < 4 = x_{(5)} < 5 = x_{(6)} = x_{(q)}.$$

Here, the number q of different endpoints is equal to $q = 6$. We then take $x_{(q+1)} = x_{(7)} = +\infty$.

Then, we use bisection to find the value k $(1 \le k \le q)$ for which the inequalities (8) and (9) hold. At each iteration of this bisection, we have an interval $[k^-, k^+]$ that is guaranteed to contain k. In the beginning, $k^- = 1$

and $k^+ = q = 6$. At the first iteration, we compute the midpoint $k_{\rm mid} = \lfloor (k^- + k^+)/2 \rfloor = \lfloor (1+6)/2 \rfloor = 3$, and check both inequalities (8) and (9) for k equal to this midpoint value, i.e., for $k = 3$.

Let us first check the inequality (8). For $k = 3$, we have $x_{(k)} = x_{(3)} = 2$, and $x_{(k+1)} = x_{(4)} = 3$. The condition $\overline{x}_j \le x_{(k)} = 2$ is only satisfied for the second interval $j = 2$ for which $\overline{x}_2 = 2$, so the left-hand side of (8) takes the form

$$\sum_{j:\overline{x}_j \le x_{(k)}} m_j \cdot (x_{(k)} - \overline{x}_j) = m_2 \cdot (x_{(3)} - \overline{x}_2) = 0.$$

The condition $\underline{x}_i \ge x_{(k+1)} = 3$ is only satisfied for the third interval $i = 3$ for which $\underline{x}_i = 3$. Thus, the right-hand side of (8) takes the form

$$\sum_{i:\underline{x}_i \ge x_{(k+1)}} m_i \cdot (\underline{x}_i - x_{(k)}) = m_3 \cdot (\underline{x}_3 - x_{(k)}) = \frac{1}{4} \cdot (3-2) = \frac{1}{4}.$$

Since $0 \le \dfrac{1}{4}$, the condition (8) is satisfied.

Similarly, the inequality (9) takes the form

$$m_2 \cdot (x_{(4)} - \overline{x}_2) > m_3 \cdot (\underline{x}_3 - x_{(4)}),$$

i.e., $\dfrac{1}{4} \cdot (3-2) > \dfrac{1}{4} \cdot (3-3)$ which is clearly true.

Since both inequalities (8) and (9) hold for his $k = 3$, this means that we have found the desired k: it is $k = 3$.

According to the algorithm, we now compute the corresponding values of S_k and Σ_k:

$$S_3 \stackrel{\rm def}{=} \sum_{j:\overline{x}_j \le x_{(3)}} m_j \cdot \overline{x}_j + \sum_{i:\underline{x}_i \ge x_{(4)}} m_i \cdot \underline{x}_i = m_2 \cdot \overline{x}_2 + m_3 \cdot \underline{x}_3 = \frac{1}{4} \cdot 2 + \frac{1}{4} \cdot 3 = \frac{5}{4}$$

and

$$\Sigma_3 \stackrel{\rm def}{=} \sum_{j:\overline{x}_j \le x_{(3)}} m_j + \sum_{i:\underline{x}_i \ge x_{(4)}} m_i = m_2 + m_3 = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}.$$

Then, we compute $r_3 = S_3/\Sigma_3 = \dfrac{5}{2}$, after which we compute

$$\underline{V} = \sum_{j:\overline{x}_j \le x_{(3)}} m_j \cdot \left(\overline{x}_j - \frac{5}{2}\right)^2 + \sum_{i:\underline{x}_i \ge x_{(4)}} m_i \cdot \left(\underline{x}_i - \frac{5}{2}\right)^2 =$$

$$m_2 \cdot \left(\overline{x}_2 - \frac{5}{2}\right)^2 + m_3 \cdot \left(\underline{x}_3 - \frac{5}{2}\right)^2 = \frac{1}{4} \cdot \left(\frac{1}{2}\right)^2 + \frac{1}{4} \cdot \left(\frac{1}{2}\right)^2 = \frac{1}{8}.$$

So, the smallest possible value of the variance – equal to $\underline{V} = \dfrac{1}{8}$ – is attained when x_1 is located at the midpoint $\dfrac{5}{2}$ of the corresponding interval, x_2 is located at the point $\overline{x}_2 = 2$, and x_3 is located at the point $\underline{x}_3 = 3$.

Computing $\overline{V}$: Stage 1. In accordance with our algorithm, first, we sort all $n = 3$ midpoints $\widetilde{x}_1 = 2.5$, $\widetilde{x}_2 = 1.5$, and $\widetilde{x}_3 = 3.5$ into a non-decreasing sequence: $1.5 < 2.5 < 3.5$. After this sorting, we can assume that the intervals $\mathbf{x}_i$ are sorted in such a way that $\widetilde{x}_1 \le \widetilde{x}_2 \le \ldots \le \widetilde{x}_n$. In other words, we now have $[\underline{x}_1, \overline{x}_1] = [1, 2]$, $[\underline{x}_2, \overline{x}_2] = [0, 5]$, and $[\underline{x}_3, \overline{x}_3] = [3, 4]$, so that

$$\widetilde{x}_1 = 1.5 < \widetilde{x}_2 = 2.5 < \widetilde{x}_3 = 3.5.$$

In the new ordering, $m_1 = \dfrac{1}{4}$, $m_2 = \dfrac{1}{2}$, and $m_3 = \dfrac{1}{4}$. We take $\widetilde{x}_{n+1} = \widetilde{x}_4 = +\infty$.

The resulting sequence of midpoints $\widetilde{x}_k$ is strictly increasing, so all the values k are proper. Since all the values k are proper, for each k, we have $s(k) = l(k) = k$.

Computing $\overline{V}$: Stage 2. Now, we use bisection to find the value k ($1 \le k \le n$) for which the following inequalities (11) and (12) hold. At each iteration of this bisection, we have an interval $[k^-, k^+]$ that is guaranteed to contain k. In the beginning, $k^- = 1$ and $k^+ = n + 1 = 4$. At the first iteration, we compute the midpoint $k_{\rm mid} = \lfloor (1 + 4)/2 \rfloor = 2$, and check both inequalities (11) and (12) for $k = 2$.

The left-hand side of (11) takes the form

$$\sum_{j=2}^{3} m_j \cdot (\overline{x}_j - \widetilde{x}_2) = m_2 \cdot (\overline{x}_2 - \widetilde{x}_2) + m_3 \cdot (\overline{x}_3 - \widetilde{x}_2) = \frac{1}{2} \cdot (5 - 2.5) + \frac{1}{4} \cdot (4 - 2.5) = \frac{13}{8},$$

while the right-hand side of (11) takes the form

$$\sum_{i=1}^{1} m_i \cdot (\widetilde{x}_2 - \underline{x}_i) = m_1 \cdot (\widetilde{x}_2 - \underline{x}_1) = \frac{1}{4} \cdot (2.5 - 1) = \frac{3}{8}.$$

Since $13/8 \not< 3/8$, the inequality (11) does not hold.

In accordance with the algorithm, we thus keep $k^+ = 4$ and replace the original value of $k^- = 1$ with the new value $k^- = k_{\rm mid} + 1 = 3$.

Now again, with the new values $k^- = 3$ and $k^+ = 4$, we again compute the midpoint $k_{\rm mid} = \lfloor (3 + 4)/2 \rfloor = 3$. For this midpoint, the left-hand side of (11) takes the form

$$\sum_{j=3}^{3} m_j \cdot (\overline{x}_j - \widetilde{x}_3) = m_3 \cdot (\overline{x}_3 - \widetilde{x}_3) = \frac{1}{4} \cdot (4 - 3.5) = \frac{1}{8},$$

while the right-hand side of (11) takes the form

$$\sum_{i=1}^{2} m_i \cdot (\widetilde{x}_3 - \underline{x}_i) = m_1 \cdot (\widetilde{x}_3 - \underline{x}_1) + m_2 \cdot (\widetilde{x}_3 - \underline{x}_2) = \frac{1}{4} \cdot (3.5 - 1) + \frac{1}{2} \cdot (3.5 - 0) = \frac{19}{8}.$$

Here, $\dfrac{1}{8} < \dfrac{19}{8}$, so the inequality (11) holds.

Let us check the inequality (12) for this $k = 3$. The left-hand side of this inequality is equal to

$$\sum_{j=3}^{3} m_j \cdot (\overline{x}_j - \widetilde{x}_2) = m_3 \cdot (\overline{x}_3 - \widetilde{x}_2) = \frac{1}{4} \cdot (4 - 2.5) = \frac{3}{8},$$

while its right-hand side is equal to

$$\sum_{i=1}^{2} m_i \cdot (\widetilde{x}_2 - \underline{x}_i) = m_1 \cdot (\widetilde{x}_2 - \underline{x}_1) + m_2 \cdot (\widetilde{x}_2 - \underline{x}_2) = \frac{1}{4} \cdot (2.5 - 1) + \frac{1}{2} \cdot (2.5 - 0) = \frac{13}{8}.$$

The equality (12) is thus not satisfied. Thus, according to our algorithm, we keep $k^- = 3$ and replace the original value $k^+ = 4$ with the new value $k^+ = k_{\rm mid} - 1 = 2$. As a result, we get an interval $[k^-, k^+] = [3, 2]$ in which the lower endpoint is smaller than the upper endpoint – this means that this interval is empty, i.e., that there is no value k which satisfies both inequalities (11) and (12).

Computing $\overline{V}$: Stage 3. In accordance with our algorithm, we now use bisection to find the proper value k ($1 \le k \le n$) for which the inequalities (13) and (14) hold. At each iteration of this bisection, we have an interval $[k^-, k^+]$ that is guaranteed to contain k. In the beginning, $k^- = 1$ and $k^+ = n = 3$. At the first iteration, we compute the midpoint $k_{\rm mid} = s(\lfloor (1+3)/2 \rfloor) = s(2) = 2$, and check both inequalities (13) and (14) for $k = 2$.

Since $l(k) = k = 2$, the left-hand side of the inequality (13) takes the value

$$\sum_{j=3}^{3} m_j \cdot (\overline{x}_j - \widetilde{x}_2) = m_3 \cdot (\overline{x}_3 - \widetilde{x}_2) = \frac{1}{4} \cdot (4 - 2.5) = \frac{3}{8},$$

while its right-hand side takes the value

$$\sum_{i=1}^{2} m_i \cdot (\widetilde{x}_2 - \underline{x}_i) = m_1 \cdot (\widetilde{x}_2 - \underline{x}_1) + m_2 \cdot (\widetilde{x}_2 - \underline{x}_2) = \frac{1}{4} \cdot (2.5 - 1) + \frac{1}{2} \cdot (2.5 - 0) = \frac{13}{8}.$$

Since $\frac{3}{8} \le \frac{13}{8}$, the inequality (13) is satisfied.

The left-hand side of the inequality (14) has the value

$$\sum_{j=2}^{3} m_j \cdot (\overline{x}_j - \widetilde{x}_2) = m_2 \cdot (\overline{x}_2 - \widetilde{x}_2) + m_3 \cdot (\overline{x}_3 - \widetilde{x}_2) = \frac{1}{2} \cdot (5 - 2.5) + \frac{1}{4} \cdot (4 - 2.5) = \frac{13}{8},$$

while its right-hand side has the value

$$\sum_{i=1}^{1} m_i \cdot (\widetilde{x}_2 - \underline{x}_i) = m_1 \cdot (\widetilde{x}_2 - \underline{x}_1) = \frac{1}{4} \cdot (2.5 - 1) = \frac{3}{8}.$$

Since $\frac{13}{8} \geq \frac{3}{8}$, the inequality (14) is also satisfied.

For $k = 2$, both inequalities are satisfied, therefore, according to the algorithm, we compute

$$V = \sum_{i=1}^{1} m_i \cdot (\underline{x}_i - \widetilde{x}_2)^2 + \sum_{j=2}^{3} m_j \cdot (\overline{x}_j - \widetilde{x}_2)^2 =$$

$$m_1 \cdot (\underline{x}_1 - \widetilde{x}_2)^2 + m_2 \cdot (\overline{x}_2 - \widetilde{x}_2)^2 + m_3 \cdot (\overline{x}_3 - \widetilde{x}_2)^2 =$$

$$\frac{1}{4} \cdot (1 - 2.5)^2 + \frac{1}{2} \cdot (5 - 2.5)^2 + \frac{1}{4} \cdot (4 - 2.5)^2 = \frac{17}{4} = 4\frac{1}{4}.$$

Computing $\overline{V}$: Final stage. In general, as $\overline{V}$, we take the largest of the two values of V obtained on the second and on the third stages. Since on the second stage, we did not get any value, we thus conclude that $\overline{V} = 4\frac{1}{4}$.

This largest value of the variance is attained when we have a probability distribution which is located:

- for the interval $[1, 2]$, at the lower endpoint 1;
- for the interval $[3, 4]$, at the upper endpoint 4;
- for the interval $[0, 5]$, at both endpoints 0 and 5 with equal probability $\frac{1}{2}$.

Conclusion. For the above example, the interval of possible values of the variance V is $[\underline{V}, \overline{V}] = \left[\frac{1}{8}, 4\frac{1}{4}\right]$.

4 Justification of the Algorithms

Let us show that the above algorithms $\underline{\mathcal{V}}$ and $\overline{\mathcal{V}}$ always compute the endpoints of the range $[\underline{V}, \overline{V}]$ in time $O(n \cdot \log(n))$.

Computing $\underline{V}$: preliminary analysis of the problem. Let us start with the problem of computing $\underline{V}$. According to [6], to find the value $\underline{V}$, we must find the values $x_i \in \mathbf{x}_i$ for which the expression $\sum\limits_{i=1}^{n} m_i \cdot x_i^2 - E^2$, where $E = \sum\limits_{i=1}^{n} m_i \cdot x_i$, is the smallest possible.

Let us start the analysis of this problem with simple calculus. Let $f(x_1, \ldots, x_n)$ be a differentiable function on a box $B \stackrel{\text{def}}{=} \mathbf{x}_1 \times \ldots \times \mathbf{x}_n$, and let $x^- = (x_1^-, \ldots, x_n^-) \in B$ be a point at which f attains its smallest value on this box.

Then, for every i, the function $f_i(x_i) \stackrel{\text{def}}{=} f(x_1^-, \ldots, x_{i-1}^-, x_i, x_{i+1}^-, \ldots, x_n^-)$ also attains its minimum on the interval $[\underline{x}_i, \overline{x}_i]$ at the point $x_i = x_i^-$.

According to the basic calculus, this minimum is either attained in the interior of the interval, in which case $df_i/dx_i = 0$ for $x_i = x_i^-$, or the minimum is attained at one of the endpoints of the interval $[\underline{x}_i, \overline{x}_i]$. If the minimum is attained at the left endpoint $\underline{x}_i$, then the function f_i cannot be decreasing at this point, so $df_i/dx_i \geq 0$. Similarly, if the minimum is attained at the right endpoint $\overline{x}_i$, then $df_i/dx_i \leq 0$.

By definition of the function $f_i(x_i)$, the value of the derivative df_i/dx_i for $x_i = x_i^-$ is equal to the value of the partial derivative $\partial f/\partial x_i$ at the point x^-. Thus, for each i, we have one of the following three cases:

- either $\underline{x}_i < x_i^- < \overline{x}_i$ and $\partial f/\partial x_i = 0$;
- or $x_i^- = \underline{x}_i$ and $\partial f/\partial x_i \geq 0$;
- or $x_i^- = \overline{x}_i$ and $\partial f/\partial x_i \leq 0$.

For $f = V$, as one can easily see, $\partial V/\partial x_i = 2m_i \cdot (x_i - E)$, so the sign of this derivative is the same as the sign of the difference $x_i - E$. Therefore, for the point x^- at which the variance V attains its minimum, we have one of the following three situations:

- either $\underline{x}_i < x_i^- < \overline{x}_i$ and $x_i^- = E$;
- or $x_i^- = \underline{x}_i$ and $x_i^- \geq E$;
- or $x_i^- = \overline{x}_i$ and $x_i^- \leq E$.

In the first case, $\underline{x}_i < E < \overline{x}_i$; in the second case, $E \leq \underline{x}_i$, and in the third case, $\overline{x}_i \leq E$.

Computing $\underline{V}$: towards an algorithm. Let us show that if we know where E is in comparison to the endpoints of all the intervals, i.e., to which "zone" $[x_{(k)}, x_{(k+1)}]$ the value E belongs, we can uniquely determine the values x_i^- for all i.

Indeed, when $x_{(k+1)} \leq \underline{x}_i$, this means that $E \leq x_{(k+1)} \leq \underline{x}_i$, so $E \leq \underline{x}_i$. Thus, we cannot have the first case (in which $E > \underline{x}_i$), so we must have either the second or the third cases, i.e., we must have $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$. If the interval $\mathbf{x}_i$ is degenerate, then both cases lead to the same result. If the interval is non-degenerate, then we cannot have the third case – in which $\underline{x}_i < \overline{x}_i \leq E$ hence $\underline{x}_i < E$ – and thus, we must have the second case, i.e., $x_i^- = \overline{x}_i$. Thus, $x_{(k+1)} \leq \underline{x}_i$ implies that $x_i^- = \underline{x}_i$.

Similarly, $x_{(k)} \geq \overline{x}_i$ implies that $x_i^- = \overline{x}_i$, and in all other cases, we have $x_i^- = E$.

All that remains is to find the appropriate k. Once k is fixed, we can find the values x_i^- in linear time, and then compute the corresponding value V in linear time. The only condition on k is that the average of the corresponding values x_i^- should be within the corresponding zone $[x_{(k)}, x_{(k+1)})$.

Computing $\underline{V}$: first algorithm. In principle, we can find k by exhaustive (linear) search. Since there are $2n$ possible small intervals, we must therefore repeat $O(n)$ computations $2n$ times, which takes $2n \cdot O(n) = O(n^2)$ time. Together with the original sorting – that takes $O(n \cdot \log(n))$ time – we thus get a quadratic time algorithm, since

$$O(n^2) + O(n \cdot \log(n)) = O(n^2).$$

Computing $\underline{V}$: towards a faster algorithm. Let us now show that we can find k faster. We want to satisfy the conditions $x^{(k)} \le E$ and $E < x^{(k+1)}$. The value E is the weighted average of all the values x_i^-, i.e., we have

$$E = S_k + (1 - \Sigma_k) \cdot E, \tag{15}$$

where S_k is defined by the formula (10) and Σ_k is defined in the description of the algorithm $\underline{\mathcal{V}}$. By moving all the terms proportional to E to the left-hand side of (15), we conclude that $\Sigma_k \cdot E = S_k$, i.e., that $E = S_k/\Sigma_k$ $(= r_k$; the case when $\Sigma_k = 0$ is handled later in this proof). The first desired inequality $x^{(k)} \le E$ thus takes the form $S_k/\Sigma_k \le x^{(k)}$, i.e., equivalently, $\Sigma_k \cdot x^{(k)} \le S_k$, i.e.,

$$\left(\sum_{i:\underline{x}_i \ge x_{(k+1)}} m_i + \sum_{j:\overline{x}_j \le x_{(k)}} m_j \right) \cdot x_{(k)} \le \sum_{i:\underline{x}_i \ge x_{(k+1)}} m_i \cdot \underline{x}_i + \sum_{j:\overline{x}_j \le x_{(k)}} m_j \cdot \overline{x}_j. \tag{16}$$

If we subtract $m_i \cdot x_{(k)}$ (or, correspondingly, $m_j \cdot x_{(k)}$) from each term in the right-hand side and move terms proportional to $\overline{x}_j - x_{(k)}$ is to the left-hand side of the inequality, we get the desired inequality (8).

When k increases, the left-hand side of the inequality (8) increases – because each term increases and new terms may appear. Similarly, the right-hand side of this inequality decreases with k. Thus, if this inequality holds for k, it should also hold for all smaller values, i.e., for $k-1$, $k-2$, etc.

Similarly, the second desired inequality $E < x^{(k+1)}$ takes the equivalent form (9). When k increases, the left-hand side of this inequality increases, while the right-hand side decreases. Thus, if this inequality is true for k, it is also true for $k+1$, $k+2$, ...

If both inequalities (8) and (9) are true for two different values $k < k'$, then they should both be true for all the values intermediate between k and k', i.e., for $k+1, k+2, \ldots, k'-1$. Let us show that both inequalities cannot be true for k and for $k+1$. Indeed, if the inequality (8) is true for $k+1$, this means that

$$\sum_{j:\overline{x}_j \le x_{(k+1)}} m_j \cdot (x_{(k+1)} - \overline{x}_j) \le \sum_{i:\underline{x}_i \ge x_{(k+2)}} m_i \cdot (\underline{x}_i - x_{(k+1)}). \tag{17}$$

However, the left-hand side of this inequality is not smaller than the left-hand side of (9), while the right-hand side of this inequality is not larger than the right-hand side of (9). Thus, (17) is inconsistent with (9). This inconsistency proves that there is only one k for which both inequalities are true, and this k can be found by the bisection method as described in the above algorithm $\underline{\mathcal{V}}$.

Computing $\underline{V}$: computation time for the resulting algorithm. How long does this algorithm take? In the beginning, we only know that k belongs to the interval $[1, 2n]$ of width $O(n)$. At each stage of the bisection step, we divide the interval (containing k) in half. After I iterations, we decrease the width of this interval by a factor of 2^I. Thus, to find the exact value of k, we must have I for which $O(n)/2^I = 1$, i.e., we need $I = O(\log(n))$ iterations. On each iteration, we need $O(n)$ steps, so we need a total of $O(n \cdot \log(n))$ steps. With $O(n \cdot \log(n))$ steps for sorting, and $O(n)$ for computing the variance, we get a $O(n \cdot \log(n))$ algorithm. The statement about the algorithm $\underline{\mathcal{V}}$ is proven.

Comment. In the above text, we considered the case when $\Sigma_k \neq 0$. In a comment after the description of the algorithm for computing $\underline{V}$, we have mentioned that it is possible to have $\Sigma_k = 0$, i.e., it is possible that for all the values i, we have $\underline{x}_i < x_{(k+1)}$ and $x_{(k)} < \overline{x}_i$.

In this case, since the values $x_{(k)}$ are sorted endpoints $\underline{x}_i$ and $\overline{x}_i$, from the fact that $\underline{x}_i < x_{(k+1)}$), we conclude that $\underline{x}_i \leq x_{(k)}$ – since $x_{(k)}$ is the largest of the endpoints which are smaller than $x_{(k+1)}$.

Similarly, $x_{(k)} < \overline{x}_i$ implies that $x_{(k+1)} \leq \overline{x}_i$. Therefore, in this case, $\underline{x}_i \leq x_{(k)} \leq x_{(k+1)} \leq \overline{x}_i$ for all i. Hence, all the intervals $\mathbf{x}_i$ contain the value $x_{(k)}$. If on each interval $\mathbf{x}_i$, we take a distribution that is located at $x_{(k)}$ with probability 1, we get the resulting 1-point distribution for which $V = 0$. Thus, in this case, indeed $\underline{V} = 0$ (in accordance with the above algorithm).

Computing $\overline{V}$: preliminary analysis. As shown in [6], the largest possible value $\overline{V}$ of the variance V is attained when for each i, the distribution ρ_i is located at two points: $\overline{x}_i$ and $\underline{x}_i$, and the value $\overline{V}$ is the maximum of the expression (3). If we denote $p_i \stackrel{\text{def}}{=} \overline{m}_i/m_i$, then we have $\overline{m}_i = m_i \cdot p_i$, $\underline{m}_i = m_i - \overline{m}_i = m_i \cdot (1 - p_i)$, and the conditions that $\underline{m}_i \geq 0$ and $\overline{m}_i \geq 0$ are equivalent to $p_i \in [0, 1]$.

In terms of the new variables p_i, to find $\overline{V}$, we must find the values $p_i \in [0, 1]$ for which the following expression attains the largest possible value:

$$V = \sum_{i=1}^{n} m_i \cdot (p_i \cdot \overline{x}_i^2 + (1 - p_i) \cdot \underline{x}_i^2) - E^2,$$

where

$$E = \sum_{i=1}^{n} m_i \cdot (p_i \cdot \overline{x}_i + (1 - p_i) \cdot \underline{x}_i).$$

Let us apply the calculus-based analysis to the above problem of maximizing the expression V as a function of n variables $p_1, \ldots, p_n$. Here,

$$\frac{\partial V}{\partial p_i} = m_i \cdot (\overline{x}_i^2 - \underline{x}_i^2) - 2 \cdot E \cdot (\overline{x}_i - \underline{x}_i) =$$

$$2m_i \cdot (\overline{x}_i - \underline{x}_i) \cdot \left(\frac{\underline{x}_i + \overline{x}_i}{2} - E \right) = 2m_i \cdot (\overline{x}_i - \underline{x}_i) \cdot (\widetilde{x}_i - E),$$

where $\widetilde{x}_i$ is the midpoint of the interval $\mathbf{x}_i$. So, the sign of this derivative coincides with the sign of the difference $\widetilde{x}_i - E$. Thus, similarly to the case of $\underline{V}$, from the fact that V attains maximum, we conclude that for every i, we have three possible situations:

- either $0 < p_i < 1$ and $\widetilde{x}_i = E$;
- or $p_i = 0$ and $\widetilde{x}_i \le E$;
- or $p_i = 1$ and $\widetilde{x}_i \ge E$.

Computing $\overline{V}$: towards an algorithm. Let us show that if we know where E is in comparison to the midpoints $\widetilde{x}_i$ of all the intervals, then we can uniquely determine almost all the values p_i – except a few with the same $\widetilde{x}_i$.

Indeed, when $\widetilde{x}_i > E$, then we cannot have neither the first case (in which $E = \underline{x}_i$) nor the second case, so we must the third case $p_i = 1$, i.e., we must have $x_i = \overline{x}_i$ with probability 1.

Similarly, when $\widetilde{x}_i < E$, then we have $p_i = 0$, i.e., we have $x_i = \underline{x}_i$ with probability 1.

When $\widetilde{x}_i = E$, then we cannot say anything about p_i: all we know is that we have $\overline{x}_i$ with some probability p_i and $\underline{x}_i$ with the probability $1 - p_i$.

In our algorithm, we have sorted the intervals in such a way that their midpoints form an increasing sequence. So, we can assume that the values $\widetilde{x}_i$ are already sorted. In principle, there are two possible cases:

- the mean value E corresponding to the optimal distribution is different from all the values $\widetilde{x}_i$, and
- the mean value E corresponding to the optimal distribution coincides with one of the values $\widetilde{x}_i$.

Let us show that both cases are indeed possible:

- If we have two intervals $[-5, -4]$ and $[4, 5]$ with probability $1/2$ each, then the mean value E must be within the interval $[(-5+4)/2, (5-4)/2] = [-0.5, 0.5]$ and therefore, cannot coincide with any of the midpoints -4.5 and 4.5.
- On the other hand, in the above-cited example where we have three intervals $[0, 1]$ with probability $1/3$ each, we must have $E = \widetilde{x}_i$ for some i, because otherwise all three distributions ρ_i would be concentrated on one of the endpoints, and we already know that this way, we cannot attain the maximum of V.

Let us analyze these two cases one by one.

Case 1. In the first case, let k denote the smallest integer for which $\widetilde{x}_k > E$. Then, according to the above description, we have $x_i = \underline{x}_i$ for $i < k$ and $x_j = \overline{x}_j$ for $j \geq k$, hence $E = \sum\limits_{i=1}^{k-1} m_i \cdot \underline{x}_i + \sum\limits_{j=k}^{n} m_j \cdot \overline{x}_j$. Our selection of k means that $\widetilde{x}_{k-1} \leq E < \widetilde{x}_k$. Substituting the expression for E into this double inequality, we get the inequalities described in the algorithm.

Similar to the proof of correctness for the algorithm $\underline{V}$, we can conclude that there is only one such k, and that the corresponding value k can indeed be found by the bisection described in the algorithm.

Case 2. In the second case, let k be the first value for which $E = \widetilde{x}_k$. By definition of k, we must have $\widetilde{x}_k > \widetilde{x}_{k-1}$, so this k is a proper value. Let us recall that for each k, by $l(k)$ we denoted the largest index for which $\widetilde{x}_{l(k)} = \widetilde{x}_k$. Then, we have

$$E = \widetilde{x}_k = \sum_{i=1}^{k-1} m_i \cdot \underline{x}_i + \sum_{i=k}^{l(k)} m_i \cdot E_i + \sum_{j=l(k)+1}^{n} m_j \cdot \overline{x}_j,$$

where by E_i, we denoted the mean of $\rho_i(x)$. Since $E_i \in [\underline{x}_i, \overline{x}_i]$, we can find the interval of possible values of the right-hand side of this expression – namely, to get the lower bound, we replace E_i with $\underline{x}_i$, and to get the upper bound, we replace E_i with the upper bound $\overline{x}_i$. Thus, we conclude that the actual value $\widetilde{x}_k$ must be between the endpoints of this interval:

$$\sum_{i=1}^{l(k)} m_i \cdot \underline{x}_i + \sum_{j=l(k)+1}^{n} m_j \cdot \overline{x}_j \leq \widetilde{x}_k \leq \sum_{i=1}^{k-1} m_i \cdot \underline{x}_i + \sum_{j=k}^{n} m_j \cdot \overline{x}_j.$$

Similarly to the proof for $\underline{V}$, we can now conclude that Part 3 of the algorithm describes how to find the corresponding value k.

We will just mention that when $\widetilde{x}_k = E$, then $(\underline{x}_i - E)^2 = (\overline{x}_i - E)^2$, hence, no matter what p_i is, the corresponding two terms

$$m_i \cdot p_i \cdot (\overline{x}_i - E)^2 + m_i \cdot (1 - p_i) \cdot (\underline{x}_i - E)^2$$

in the expression for the variance always add up to the same value $m_i \cdot (\overline{x}_i - E)^2$.

The algorithm has been justified.

5 Beyond Mean and Variance: A Comment

Beyond mean, to arbitrary monotonic statistical characteristics. As proven in [6], for the mean E:

- the smallest possible value of E is attained when on each interval $[\underline{x}_i, \overline{x}_i]$, the entire mass is concentrated on the lower endpoint $\underline{x}_i$, i.e., we have $\underline{x}_i$ with probability m_i, and

- the largest possible value of E is attained when on each interval $[\underline{x}_i, \overline{x}_i]$, the mass is concentrated on the lower endpoint $\underline{x}_i$, i.e., we have $\overline{x}_i$ with probability m_i.

Comment. If, for several intervals $\mathbf{x}_i$, their lower endpoints $\underline{x}_i$ coincide, then, of course, we have to add the corresponding probabilities m_i to describe the probability of the corresponding lower endpoint; same for upper endpoints.

One can easily see that the same is true for all statistical characteristics which are monotonic in the sense of *stochastic dominance* – a natural generalization of standard order to probability distributions.

- If we know the exact values x and y of two variables, then we can say that y *dominates* x if $y \geq x$.
- If x and y are random variables, then it is natural to say that y *dominates* x if for every real number t, the probability that y exceeds t is larger than (or equal to) the probability that x exceeds t.

The probability $\text{Prob}(x > t)$ that $x > t$ can be described as $1 - F_x(t)$, where $F_x(t) \stackrel{\text{def}}{=} \text{Prob}(x \leq t)$ is the corresponding value of the cumulative distribution function (cdf). Thus, the condition that $1 - F_y(t) \geq 1 - F_x(t)$ can be reformulated as $F_y(t) \leq F_x(t)$. So, a probability distribution with a cumulative distribution function $F_y(t)$ is said to *dominate* a probability distribution with a cumulative distribution function $F_x(t)$ if $F_y(t) \leq F_x(t)$ for every real number t. A statistical characteristic $C(\rho)$ is called *monotonic* if $C(\rho) \geq C(\rho')$ whenever the distribution described by the density ρ dominates the distribution described by the density ρ'.

Mean is a monotonic characteristic; another monotonic characteristic is the *median*, i.e., the value m for which $F(m) = 1/2$.

Similarly to the mean, for every monotonic statistical characteristic C,

- the smallest possible value of C is attained when on each interval $[\underline{x}_i, \overline{x}_i]$, the entire mass is concentrated on the lower endpoint $\underline{x}_i$, i.e., we have $\underline{x}_i$ with probability m_i, and
- the largest possible value of C is attained when on each interval $[\underline{x}_i, \overline{x}_i]$, the mass is concentrated on the lower endpoint $\underline{x}_i$, i.e., we have $\overline{x}_i$ with probability m_i.

Thus, it is easy to compute the range $[\underline{C}, \overline{C}]$ of a monotonic characteristic under the Dempster-Shafer uncertainty.

Beyond variance. Similar algorithms can be described not only for the variance, but also for the characteristic $C = E + k_0 \cdot \sigma$ (where $\sigma = \sqrt{V}$ and k_0 is a fixed number), a characteristic which is useful in describing confidence intervals and outliers; see, e.g., [5, 8].

For C, the proposition from [6] is still true: indeed, replacing two points with their mean decreases σ and leaves E intact, hence decreases C as well. Thus, in this case, the minimum of C is also attained for 1-point distributions; so we can use a natural generalization of interval algorithms from [5] to describe this more general case as well.

For C, the maximum is also attained for two-point distributions. Differentiating the resulting expression for C w.r.t. p_i, we conclude that the sign of the derivative coincides with the sign of the difference $\widetilde{x}_i - E'$ for some linear combination E' of E and σ. So, once we know where E' is in relation to the midpoints, we can make a similar conclusion about the maximizing distributions ρ_i – the only difference is that now the formulas expressing E' in terms of the selected values x_i are more complex.

6 Conclusions

In many real-life situations, we only have partial information about the actual probability distribution. For example, under Dempster-Shafer uncertainty, we only know the masses $m_1, \ldots, m_n$ assigned to different sets $S_1, \ldots, S_n$, but we do not know the distribution within each set S_i. Because of this uncertainty, there are many possible probability distributions consistent with our knowledge; different distributions have, in general, different values of standard statistical characteristics such as mean and variance. It is therefore desirable, given a Dempster-Shafer knowledge base, to compute the *ranges* of possible values of mean and of variance.

In their recent paper, A. T. Langewisch and F. F. Choobineh show how to compute these ranges in polynomial time. In particular, they reduce the problem of computing $\overline{V}$ to the problem of minimizing a convex quadratic function, a problem which can be solved in time $O(n^2 \cdot \log(n))$. We show that the corresponding quadratic optimization problem can be actually solved in time $O(n \cdot \log(n))$; thus, we can compute the bounds $\underline{V}$ and $\overline{V}$ in time $O(n \cdot \log(n))$.

It is worth mentioning that while for the Dempster-Shafer uncertainty, there exist efficient algorithms for computing the range of the variance, in a similar situation of interval uncertainty, the problem of computing the range for variance is NP-hard. Thus, with respect to computing the values (and ranges) of statistical characteristics, the case of Dempster-Shafer uncertainty is computationally simpler than the case of interval uncertainty.

Acknowledgments

This work was supported in part by NASA under cooperative agreement NCC5-209, by NSF grant EAR-0225670, by NIH grant 3T34GM008048-20S1, and by Army Research Lab grant DATM-05-02-C-0046. The authors are greatly thankful to the anonymous referees for their great help.

References

[1] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.

[2] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, Computing Variance for Interval Data is NP-Hard, *ACM SIGACT News*, 2002, 33(2):108–118.

[3] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, Exact Bounds on Finite Populations of Interval Data, *Reliable Computing*, 2005, 11(3):207–233.

[4] 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, editors, *Numerical Software with Result Verification*, Proceedings of the International Dagstuhl Seminar, Dagstuhl Castle, Germany, January 19–24, 2003, Springer Lectures Notes in Computer Science, 2004, Vol. 2991, pp. 274–305.

[5] V. Kreinovich, L. Longpré, P. Patangay, S. Ferson, and L. Ginzburg, Outlier Detection Under Interval Uncertainty: Algorithmic Solvability and Computational Complexity, Reliable Computing, 2005, 11(1):59–76.

[6] A. T. Langewisch and F. F. Choobineh, Mean and variance bounds and propagation of uncertainty for ill-specified random variables, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 2004, 34(4):494–506.

[7] S. A. Vavasis, *Nonlinear Optimization: Complexity Issues*, Oxford University Press, New York, 1991.

[8] H. M. Wadsworth, Jr. (ed.), *Handbook of statistical methods for engineers and scientists*, McGraw-Hill Publishing Co., N.Y., 1990.

[9] R. R. Yager, J. Kacprzyk, and M. Pedrizzi (Eds.), *Advances in the Dempster-Shafer Theory of Evidence*, Wiley, N.Y., 1994

Appendix: A Simple Numerical Example where Dempster-Shafer Bounds on Variance are Different from the Interval Bounds

Let us give an example explaining that for variance, the interval range is, in general, different from the DS range corresponding to the case $m_1 = \ldots = m_n = \frac{1}{n}$. In this example, we take $n = 3$ and $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}_3 = [0, 1]$.

In the DS approach, it is possible that on each of these intervals, we have a distribution that is located on each endpoint with probability $\frac{1}{2}$. In this case, we attain the variance $V = \dfrac{1}{4}$ – the largest possible variance that we can attain for any probability distribution located on the interval $[0,1]$.

If on each interval, we pick the same value $\dfrac{1}{2}$ with probability 1, then the variance is 0. Since the variance is always non-negative, we conclude that, in the DS approach, $[\underline{V}, \overline{V}] = \left[0, \dfrac{1}{4}\right]$.

Let us now estimate the corresponding interval range. Since the sample variance is a non-negative quadratic function, its maximum is attained when each of the variables takes one of the extreme values $\underline{x}_i = 0$ or $\overline{x}_i = 1$. Out of possible combinations, the sample variance attains its largest value when the values of x_i are different, i.e., when two values coincide with 0 or 1, and the third value is equal to, correspondingly, 1 or 0. In this case, the largest possible value of sample variance is

$$V = \frac{1}{2} \cdot \left(\left(\frac{2}{3}\right)^2 + 2 \cdot \left(\frac{1}{3}\right)^2 \right) = \frac{1}{2} \cdot \frac{6}{9} = \frac{1}{3},$$

which is larger than $\dfrac{1}{4}$.