Range Estimation under Constraints is Computable Unless There Is a Discontinuity

Martine Ceberio, Olga Kosheleva, and Vladik Kreinovich

University of Texas at El Paso, El Paso, TX 79968, USA
{mceberio,olgak,vladik}@utep.edu

Abstract. One of the main problems of interval computations is computing the range of a given function over given intervals. It is known that there is a general algorithm for computing the range of computable functions over computable intervals. However, if we take into account that often in practice, not all possible combinations of the inputs are possible (i.e., that there are constraints), then it becomes impossible to have an algorithm which would always compute this range. In this paper, we explain that the main reason why range estimation under constraints is not always computable is that constraints may introduce discontinuity – and all computable functions are continuous. Specifically, we show that if we restrict ourselves to computably continuous constraints, the problem of range estimation under constraints remains computable.

1 Need for Range Estimation under Constraints

Need for data processing. To make a decision, in particular, to select an engineering design and/or control strategy, we need to know the effects of selecting different alternatives. In most engineering problems, we know how different quantities depend on each other and how they change with time. In particular, we usually know how the quantity y describing the effect depends on the values of the quantities $x_1, \ldots, x_n$ describing the decision and the surrounding environment: $y = f(x_1, \ldots, x_n)$. The resulting computations are known as *data processing*.

Need to take uncertainty into account. In the ideal situation, when we know the exact values $x_1, \ldots, x_n$ of the corresponding parameters, we can simply substitute these values into a known function f, and get the desired value y. In practice, the values $x_1, \ldots, x_n$ come from measurements, and measurements are never absolutely accurate. As a result, the measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ are, in general, somewhat different from the actual (unknown) values $x_1, \ldots, x_n$ of the corresponding quantity. Thus, the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is, in general, different from the desired value $y = f(x_1, \ldots, x_n)$. To make an appropriate decision, it is important to know how big can be the difference $\widetilde{y} - y$.

Need for range estimation. In many practical situations, the only information that we have about the measurement error $\widetilde{x}_i - x_i$ of each corresponding measurements is the upper bound Δ_i provided by the manufacturer. In this case, based on the measurement result $\widetilde{x}_i$, the only information that we can conclude about x_i is that x_i belongs to the *interval* $[\underline{x}_i, \overline{x}_i] \stackrel{\text{def}}{=} [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.

Another case of such an interval uncertainty is when the parameter x_i characterizes a manufactured part; in this case, we know that the corresponding value must lie within the tolerance interval – the interval $[\underline{x}_i, \overline{x}_i]$ within which the manufacturer of this part was required to keep this value.

Different values x_i from the corresponding intervals $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ lead, in general, to different values of $y = f(x_1, \ldots, x_n)$. It is therefore important to estimate the range of all such values, i.e., the set

$$\{f(x_1, \ldots, x_n) : x_i \in [\underline{x}_i, \overline{x}_i] \text{ for all } i\}.$$

In the usual case of continuous functions f, this range is an interval; we will denote this interval by $[\underline{y}, \overline{y}]$. Estimation of this range interval is known as *interval computations*; see, e.g., [4].

Range estimation problems are, in general, computable. It is known that for computable functions f on computable intervals $[\underline{x}_i, \overline{x}_i]$, there is an algorithm which computes the range of the given function on given intervals; see, e.g., [3].

In general, the corresponding computational problem is NP-hard (meaning that these computations may take a very long time), but there are many situations where feasible algorithms are possible for exact computations – and there are also many feasible algorithms for providing enclosures for the desired ranges; see, e.g., [3].

Need to take constraints into account. The above formulation of range estimation problem assumes that the quantities x_1, …, x_n are independent – in the sense that the set of possible values of, e.g., x_1, does not depend on the actual values of all other quantities. In practice, we often have additional *constraints* which limit possible combinations of values $(x_1, \ldots, x_n)$.

For example, if x_1 and x_2 represent the control values are two consequent moments of time, then usually, due to hardware limitations, these values cannot differ much, we should have a constraint $|x_1 - x_2| < \delta$ for some small value $\delta > 0$. In this case, instead of the range of all possible values of $f(x_1, \ldots, x_n)$ when each x_i is in the corresponding interval, we are only interested in the range of the values corresponding to the tuples $(x_1, \ldots, x_n)$ that satisfy all the known constraints.

Constraints make the problem of range estimation more complex. Adding constraints immediately makes the problem much more complex; see, e.g., [1].

What we do in this paper. In this paper, we explain that the main reason why range estimation under constraints is not always computable is that constraints may introduce discontinuity – and all computable functions are continuous. Specifically, we show that if we restrict ourselves to computable continuous constraints, the problem of range estimation under constraints remains computable.

2 Known Results: Brief Reminder

Definition 1.

- *A real number x is called* computable *if there exists an algorithm that, given a natural number k, returns a rational number r_k for which $|r_k - x| \le 2^{-k}$.*
- *An interval $[\underline{x}, \overline{x}]$ is called* computable *if both its endpoints are computable.*
- *A function $f(x_1, \ldots, x_n)$ from real numbers to real numbers is called* computable *if there exist two algorithms:*
  - *an algorithm that, given rational numbers $r_1, \ldots, r_n$, and an integer k, returns a rational number r for which $|r - f(r_1, \ldots, r_n)| \le 2^{-k}$; and*
  - *an algorithm that, given a rational number $\varepsilon > 0$, returns a rational number $\delta > 0$ such that if $|x_i - x'_i| \le \delta$ for all i, then*

$$|f(x_1, \ldots, x_n) - f(x'_1, \ldots, x'_n)| \le \varepsilon.$$

Proposition 1. [3, 5] *There exists an algorithm that, given a computable function $f(x_1, \ldots, x_n)$ and computable intervals $[\underline{x}_i, \overline{x}_i]$ $(1 \le i \le n)$, returns the range $[\underline{y}, \overline{y}]$ of this function on the given intervals.*

Proof. To compute $\overline{y}$ with a given accuracy $\varepsilon > 0$, we first use the second algorithm from the definition of a computable function to find $\delta > 0$ for which $|x_i - x'_i| \le \delta$ implies that the values of f are $(\varepsilon/2)$-close to each other. On each interval $[\underline{x}_i, \overline{x}_i]$, we then select finitely many points $\underline{x}_i$, $\underline{x}_i + \delta$, $\underline{x}_i + 2\delta$, ... After that, for each combination $(s_1, \ldots, s_n)$ of the selected points, we use the first algorithm to produce a rational number r which is $(\varepsilon/2)$-close to the corresponding value $f(s_1, \ldots, s_n)$. Our claim is that the largest $\overline{r}$ of these rational numbers is the desired ε-approximation to $\overline{y}$.

Indeed, on the one hand, each rational value r is bounded by $f(s_1, \ldots, s_n) + \dfrac{\varepsilon}{2}$. Thus, from $f(s_1, \ldots, s_n) \le \overline{y}$, we conclude that $r \le \overline{y} + \dfrac{\varepsilon}{2}$. In particular, this is true for the largest of these numbers, hence $\overline{r} \le \overline{y} + \dfrac{\varepsilon}{2}$.

On the other hand, let us consider the values x_i at which the function f attains its largest possible value $\overline{y}$: $f(x_1, \ldots, x_n) = \overline{y}$. Each value x_i from the corresponding interval is δ-close to one of the selected points s_i. Thus, each combination $(x_1, \ldots, x_n)$ is δ-close to the corresponding combination $(s_1, \ldots, s_n)$ of selected points – which, due to the choice of δ, implies that

$$|f(s_1, \ldots, s_n) - f(x_1, \ldots, x_n)| \le \frac{\varepsilon}{2}.$$

So, $f(s_1, \ldots, s_n) \ge \overline{y} - \dfrac{\varepsilon}{2}$. For the corresponding number r, we have $r \ge f(s_1, \ldots, s_n) - \dfrac{\varepsilon}{2}$ and hence, $r \ge \overline{y} - \varepsilon$. Since $\overline{r}$ is the largest of these rational numbers, we get $\overline{r} \ge r$ and therefore, $\overline{r} \ge \overline{y} - \varepsilon$.

A similar proof shows that the smallest $\underline{r}$ of the corresponding rational numbers r is an ε-approximation to $\underline{y}$. The proposition is proven.

Definition 2.

– *By a* computable constraint, *we mean a constraint of one of the following types:* $g_j(x_1,\ldots,x_n)=c_j$, $g_j(x_1,\ldots,x_n)\le c_j$, $c_j\le g_j(x_1,\ldots,x_n)$, *or* $\underline{c}_j\le g_j(x_1,\ldots,x_n)\le \overline{c}_j$, *where* $g_j(x_1,\ldots,x_n)$ *is a computable function and* c_j, $\underline{c}_j$, *and* $\overline{c}_j$ *are computable numbers.*
– *By a problem of* range estimation under constraints, *we mean the following problem:*
  - *given a computable function* $f(x_1,\ldots,x_n)$, n *computable intervals* $[\underline{x}_i,\overline{x}_i]$, *and a finite list of computable constraints,*
  - *compute the largest* $\overline{y}$ *and the smallest* $\underline{y}$ *values of* $f(x_1,\ldots,x_n)$ *for all the tuples* $(x_1,\ldots,x_n)$ *of values* $x_i\in[\underline{x}_i,\overline{x}_i]$ *which satisfy all the given constraints.*

Proposition 2. *No algorithm is possible which solves all the problems of range estimation under constraints.*

Comment. In other words, it is not possible to have an algorithm that, given the function f, the intervals $[\underline{x}_i,\overline{x}_i]$, and the constraints, would always compute the values $\underline{y}$ and $\overline{y}$.

Proof. Let us take $n=1$, $f(x_1)=x_1$, and a constraint $g(x_1)=c_1$, where $g(x_1)=\min(x_1,\max(0,x_1-1))$. One can check that for $x_1\le 0$, we get $g(x_1)=x_1$; for $0\le x_1\le 1$, we get $g(x_1)=0$, and for $x_1\ge 1$, we get $g(x_1)=x_1-1$. So, for $c_1<0$, the constraint is only satisfied for the value $x_1=c_1$, so we get $\overline{y}=c_1$; on the other hand, for $c_1=0$, the constraint $g(x_1)=c_1=0$ is satisfied for all $x_1\in[0,1]$, so we get $\overline{y}=1$. When $c_1\to 0$, the dependence of $\overline{y}$ on c_1 is discontinuous, and all computable functions are continuous; see, e.g., [5]. The proposition is proven.

3 New Result: Discontinuity Is the Only Obstacle to Computing $\underline{y}$ and $\overline{y}$

Definition 3.

– *Let the computable intervals* $[\underline{x}_i,\overline{x}_i]$ *be given, and let the computable functions* $g_1(x_1,\ldots,x_n)$, $\ldots$ *be given, and for each of these functions, let a type of the corresponding constraint be given (i.e.,* $=c_j$, $\le c_j$, $\ge c_j$, *or* $\underline{c}_j\le\cdot\le\overline{c}_j$*).*
– *For each combination* c *of the threshold values* c_j, $\underline{c}_j$, *and/or* $\overline{c}_j$, *by* $S(c)$, *we denote the set of all the tuples* $x_i\in[\underline{x}_i,\overline{x}_i]$ *which satisfy all the corresponding constraints.*
– *For each* $\delta>0$, *we say that the combinations* c *and* c' *are* δ-close *if the corresponding threshold are* δ*-close (e.g.,* $|c_j-c'_j|\le\delta$*).*
– *We say that the set of constraints is* computably continuous *if there exists an algorithm that, given a rational number* $\varepsilon>0$, *returns a rational number* $\delta>0$ *such that when* c *and* c' *are* δ*-close, then*

$d_H(S(c), S(c')) \le \varepsilon$, *where* $d_H(A, B)$ *is the Hausdorff distance* $d_H(A, B) \stackrel{\text{def}}{=} \max\left(\sup_{a\in A} d(a, B), \sup_{b\in B} d(b, A)\right)$ *and* $d(a, B) \stackrel{\text{def}}{=} \inf_{b\in B} d(a, b)$.

Proposition 3. *There exists an algorithm which solves the problem of range estimation under constraints for all computably continuous constraints.*

Comment. In other words, this algorithm, given the function $f(x_1, \ldots, x_n)$, the intervals $[\underline{x}_i, \overline{x}_i]$, and the constraints, returns the corresponding values $\underline{y}$ and $\overline{y}$.

Proof. To estimate $\underline{y}$ and $\overline{y}$ with accuracy ε, let us find $\delta > 0$ for which $|x_i - x'_i| \le \delta$ implies that the f-values are ε-close. One can then show that if $d_H(S, S') \le \delta$, then $\max_{x\in S} f(x)$ and $\max_{x\in S'} f(x)$ are ε-close [2].

For this $\delta > 0$, we can find $\beta > 0$ for which if c and c' are β-close, then $d_H(S(c), S(c')) \le \delta$. We can now replace each equality $g_j = c_j$ with inequalities $\underline{c}_j \le g_j \le \overline{c}_j$ and, as long as $|\underline{c}_j - c_j| \le \beta$ and $|\overline{c}_j - c_j| \le \beta$, we still have a δ-close set $S(c)$. The box $[\underline{x}_1, \overline{x}_1] \times \ldots$ is a computable compact set (see [1, 3, 5]), so due to the known properties of such sets, there exists β-close values c' for which the set $S(c')$ is a computable compact – and for which, therefore, the maximum $\overline{y}'$ and the minimum $\underline{y}'$ of the computable function $f(x)$ over $S(c')$ are computable. Since $S(c')$ is δ-close to $S(c)$, we have $|\overline{y}' - \overline{y}| \le \varepsilon$ and $|\underline{y}' - \underline{y}| \le \varepsilon$. The proposition is proven.

Proposition 4. *When all constraints are inequalities, with* $\underline{c}_j < \overline{c}_j$*, then we can solve all problems of range estimation for which the dependence* $S(c)$ *is continuous (not necessarily computably continuous).*

Proof. For $\beta = 2^{-k}$, $k = 0, 1, \ldots$, we estimate the ranges $[\underline{y}'_j, \overline{y}'_j]$ and $[\underline{y}''_j, \overline{y}''_j]$ of f over an inner β-approximation $S(c')$ and the outer β-approximations $S(c'')$. Then $\underline{y}'' \le \underline{y} \le \underline{y}'$ (and $\overline{y}' \le \overline{y} \le \overline{y}''$). Due to continuity, the sets $S(c')$ and $S(c'')$ will eventually become δ-close and thus, the estimates $\underline{y}'$ and $\underline{y}''$ become ε-close; when this happens, we return $\underline{y}'$ and $\overline{y}'$ as the desired ε-approximations to $\underline{y}$ and $\overline{y}$.

Acknowledgments. This work was supported in part by the National Science Foundation grants HRD-0734825, HRD-124212, and DUE-0926721.

References

1. M. Ceberio and V. Kreinovich (eds.), *Constraint Programming and Decision Making*, Springer Verlag, Berlin, Heidelberg, 2014.
2. V. Kreinovich and B. Kubica, “From computing sets of optima, Pareto sets, and sets of Nash equilibria to general decision-related set computations”, *Journal of Universal Computer Science*, 2010, Vol. 16, No. 18, pp. 2657–2685.
3. V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational complexity and feasibility of data processing and interval computations*, Kluwer, Dordrecht, 1997.
4. R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM Press, 2009.
5. K. Weihrauch, *Computable Analysis*, Springer-Verlag, Berlin, 2000.