Adding Constraints – A (Seemingly Counterintuitive but) Useful Heuristic in Solving Difficult Problems

Olga Kosheleva, Martine Ceberio,
and Vladik Kreinovich

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

1. Commonsense Intuition: The More Constraints, The More Difficult The Problem

- If we want to hire a lecturer in Computer Science, this is reasonably easy.
- However, once we impose constraints on research record etc., hiring becomes complicated.
- If a person coming to a conference is looking for a hotel to stay, this is usually an easy problem to solve.
- But once you add constraints on how far this hotel is from the conference site, the problem becomes difficult.
- Similarly, in numerical computations,
  - unconstrained optimization problems are usually reasonably straightforward to solve, but
  - once we add constraints, the problems often become much more difficult.

2. Sometimes Constraints Help: A Seemingly Counterintuitive Phenomenon

- Mathematicians often aim for an optimal control or an optimal design.
- To a practitioner, this may seem like a waste of time: once we are within ε of the maximum, we can stop.
- However, algorithmically, it is often easier to find x s.t. $f(x) \geq f_0$ by finding $x_{\max}$ s.t. $f'(x_{\max}) = 0$.
- A challenging theorem often becomes proven when we look for proofs of a more general result.
- In physics, equations were found when additional beauty constraints were imposed (Einstein, Bolzmann).
- In art, many great objects were designed within strict requirements on shape, form, etc.
- How to explain this counter-intuitive phenomenon?

3. Analysis of the Problem

- By definition:
  - when we impose an additional constraint,
  - some alternatives which were originally solutions stop being solutions –
  - since we impose extra constraints, constraints that are not always satisfied by all original solutions.
- Thus, the effect of adding a constraint is that the number of solution decreases.
- At the extreme, when we have added the largest possible number of constraints, we get a unique solution.
- It turns out that this indeed explains why adding constraints can make the problems easier.

4. Related Known Results: The Fewer Solutions, the Easier to Solve the Problem

- Many numerical problems are, in general, algorithmically undecidable:
  - no algorithm can always find a solution to an algorithmically defined system of equation;
  - no algorithm can always find a location of the maximum of an algorithmically defined function, etc.
- The proofs of most algorithmic non-computability results essentially use:
  - functions which have several maxima,
  - equations which have several solutions, etc.
- It turned out that this is not an accident: uniqueness actually implies algorithmic computability.

5. Uniqueness Implies Algorithmic Computability

- This result was applied to design many algorithms:
  - optimal approximation of functions;
  - reconstructing a convex body from its internal metric;
  - constructing a shortest path in a curved space, etc.
- On the other hand, it was proven that:
  - a general algorithm is not possible for functions that have exactly two global maxima;
  - a general algorithm is not possible for systems that have exactly two solutions.
- Moreover, there are results showing that for every m:
  - problems with exactly m solutions are, in general, more computationally difficult
  - than problems with $m-1$ solutions.

6. Resulting Recommendation: Applied Math

- The above discussion leads to the following seemingly counter-intuitive recommendation:
  - if a problem turns out to be too complex to solve,
  - maybe a good heuristic is to add constraints and make it more complex.
- For example:
  - if it is difficult to solve an applied mathematical problem,
  - maybe a good idea is not to simplify this problem but rather to make it more realistic.
- Indeed, applied mathematicians know that often,
  - learning more about the physical or engineering problem
  - helps to solve this problem.

7. Resulting Recommendation: Education

- This can also be applied to education:
  - if students have a hard time solving a class of problems,
  - maybe a good idea is not to make these problems easier, but to make them more complex.
- This may sound counter-intuitive.
- However, in pedagogy, it is a known fact:
  - if a school is failing,
  - the solution is usually not to make classes easier – this will lead to a further decline in knowledge;
  - a turnaround often happens when a new teacher starts giving challenging problems to students.
- This is in line with a general American idea – that to be satisfying, the job must be a challenge.

8. Caution

- Of course, it is important:
  - not to introduce so many constraints –
  - because then, the problem simply stops having solutions at all.
- It is difficult to guess which level of constraints will lead to inconsistency.
- Thus, it may be a good idea:
  - to simultaneously try to solve several different versions of the original problem,
  - with different number of constraints added.
- This way, we will hopefully be able to successfully solve one of these versions.

9. Acknowledgments

This work was supported in part:

- by National Science Foundation grants HRD-0734825, EAR-0225670, and DMS-0532645 and
- by Grant 1 T36 GM078000-01 from the National Institutes of Health.

Commonsense . . .
Sometimes . . .
Analysis of the Problem
Related Known . . .
Uniqueness Implies . . .
Resulting . . .
Resulting . . .
Caution
Acknowledgments
Home Page
Title Page
Page 10 of 12
Go Back
Full Screen
Close
Quit

10. Proof: Main Idea

- To compute x_0 s.t. $f(x_0) = 0$ with accuracy $\varepsilon > 0$, take an $(\varepsilon/4)$-net $\{x_1, \ldots, x_n\} \subseteq K$.
- For each i, we can compute $\varepsilon' \in (\varepsilon/4, \varepsilon/2)$ for which $B_i \stackrel{\text{def}}{=} \{x : d(x, x_i) \le \varepsilon'\}$ is a computable compact set.
- Thus, we can compute $m_i \stackrel{\text{def}}{=} \min\{|f(x)| : x \in B_i\}$.
- If $m_i = 0$, then $\exists x \left(f(x) = 0 \,\&\, d(x, x_i) < \dfrac{\varepsilon}{2}\right)$, hence $d(x_i, x_0) \le \dfrac{\varepsilon}{2}$.
- So, if $m_i = 0$ and $m_j = 0$ then

$$d(x_i, x_j) \le d(x_i, x_0) + d(x_0, x_j) \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

- Vice versa, if $d(x_i, x_0) > \varepsilon > 0$, we get $m_i > 0$.

11. Proof (cont-d)

- *Reminder:* if $d(x_i, x_0) > \varepsilon/2$, then $m_i > 0$; so:
  - if we compute all m_i with accuracy
  $$2^{-N} \le \min\{m_i : m_i > 0\},$$
  - and exclude all i with $m_i > 0$,
  - we get $d(x_i, x_0) \le \varepsilon/2$ for all remaining i.
- Thus, for all remaining i and j, we have
$$d(x_i, x_j) \le d(x_i, x_0) + d(x_0, x_j) \le \varepsilon/2 + \varepsilon/2 = \varepsilon.$$
- Then, $d(x_0, x_i) \le \varepsilon/2$ for each remaining i.
- *Minor problem:* do not know N a priori.
- *Solution:* we repeat computations for $N = 1, 2, \ldots$ until we get $d(x_i, x_j) \le \varepsilon$ for all remaining i and j.
- $f(x) = \max\limits_y f(y) \Leftrightarrow g(x) \stackrel{\text{def}}{=} f(x) - \max\limits_y f(y) = 0.$