Optimization under Fuzzy Constraints: From a Heuristic Algorithm to an Algorithm That Always Converges

Vladik Kreinovich[1] and Juan Carlos Figueroa-García[2]

[1]University of Texas at El Paso, El Paso, TX 79968, USA
vladik@utep.edu
[2]Universidad Distrital, Bogotá D.C, Colombia
jcfigueroag@udistrital.edu.co

1. Need to Select the Best Alternative

- In many practical situations, we want to select the best of the possible alternatives x.
- We want to use mathematical and computational techniques in solving such problems.
- So, we need describe this problem in precise terms.
- For this, we need to describe:
  - what is meant by "the best", and
  - what is meant by "possible alternatives".

2. What Does “the Best” Mean

- “The best” can usually be described in numerical form: we have an objective function $f(x)$ such that:
  - the larger the value of this function,
  - the better the alternative.
- For example, in economics problems, we want to maximize profit.
- In some cases, the better alternative corresponds to the smallest value of the corresponding function $g(x)$.
- For example, in economics, we may want to minimize the cost $g(x)$.
- In transportation problems, we may want to minimize travel time $g(x)$.
- Such problems can be easily reformulated in the maximization terms if we take $f(x) \stackrel{\text{def}}{=} -g(x)$.

3. The Objective Function Is Usually Continuous, Even Smooth

- Tiny changes in the selected alternative usually do not change the output much.
- So we expect that the values of the objective function $f(x)$ should not change much either.
- Thus, we expect the objective function to be continuous, or:
  - if we interpret "not much" as bounded by a certain constant times Δx,
  - as Lipschitz continuous, and thus, as differentiable almost everywhere.
- In practice, the objective function is usually smooth.

4. Which Alternatives Are Possible?

- An alternative is possible if it satisfies certain constraints.
- Usually, these constraints are equalities $g_i(x) = 0$ or inequalities $h_j(x) \ge 0$.
- For example, in chemical manufacturing, constraints are that:
  - the amount of potential pollutants $p(x)$ released into the environment
  - does not exceed some threshold t: $p(x) \le t$.
- This constraint can be described as $h_1(x) \ge 0$, where $h_1(x) \stackrel{\text{def}}{=} t - p(x)$.
- In principle, we can have more general constraints.
- Let us denote the set of all possible alternatives by $S \subset R^N$ for an appropriate N.

5. It Is Reasonable to Assume that Constraints Describe a Closed Set

- Selecting an alternative means selecting the parameters that describe this alternative.
- For example, in control applications, we select the values of the control parameters.
- For a car, we can select the acceleration and the torque, etc.
- In practice, we can set up the desired values only with some accuracy.
- Also, we can only measure how well we have set them with some accuracy.

6. Closed Set (cont-d)

- As a result, if we have a sequence of possible alternative $x_1, x_2, \ldots$ that converges to a limit alternative x, then:
  - for any desired implementation and/or measuring accuracy $\varepsilon > 0$,
  - there is a possible state x_n which is ε-close to x and
  - is, thus, practically indistinguishable from the alternative x.
- No matter how much we increase our accuracy, we cannot distinguish:
  - the limit alternative x from
  - possible alternatives.
- So, it makes sense to assume that the limit alternative x is also possible.

7. Closed Set (cont-d)

- Under this assumption, the set S of all possible alternatives has the property that:
  - if $x_i \in S$ for all i and $x_i \to x$,
  - then $x \in S$.
- In mathematical terms, this means that the set S is *closed*.

8. Comment

- This closeness assumption is the main reason why in traditional optimization problems:
  - we consider constraints of the non-strict inequality type $h_j(x) \geq 0$
  - but not constraints of the strict inequality type $h_j(x) > 0$.
- Indeed, non-strict inequalities are preserved in the limit.
- On the other hand, strict inequalities are not necessarily preserved: e.g., $2^{-i} > 0$, $2^{-i} \to 0$, but $0 \not> 0$.

9. In Practice, the Set of Possible Alternatives is Always Bounded

- In practice, the values of all the quantities are bounded; for example:
  - the speeds are limited by the speed of light,
  - the distances for Earth travel are bounded by the Earth's size,
  - accelerations are bounded by our technical abilities, etc.
- Thus, in practice, the set S of possible alternatives is always bounded.

10. Mathematical Conclusion: the Set of Possible Alternatives is a Compact Set

- Since the set S is closed and bounded, it is a compact set.
- This means, in particular, that:
  - for every continuous function $F(x)$ on this set,
  - there exists an alternative $x_{\rm opt}$ at which this function attains its maximum, i.e., at which

$$F(x_{\rm opt}) = \max_{x \in S} F(x).$$

11. Resulting Formulation: Optimization under Constraints

- Thus, the above practical problem takes the following form:
  - maximize the objective function $f(x)$ under the constraint that
    $$g_1(x) = 0, \ldots, g_m(x) = 0, h_1(x) \ge 0, \ldots, h_m(x) \ge 0$$
  - or, more generally, that $x \in S$ for some compact set S.
- Since the set S of possible alternatives is compact:
  - for continuous objective functions $f(x)$,
  - there is always an alternative x_{opt} that solves this problem.

12. Algorithms for Optimization under Constraints

- There exist many efficient algorithms for optimization under constraints.
- The most well-known methods are based on Lagrange multiplier techniques, according to which:
  - maximizing a function $f(x)$ under the constraints $g_1(x) = 0, \ldots, g_m(x) = 0$
  - can be reduced to the unconstrained problem of maximizing the auxiliary function

$$f(x) + \lambda_1 \cdot g_1(x) + \ldots + \lambda_m \cdot g_m(x).$$

- Here, λ_i are auxiliary constants (known as *Lagrange multipliers*).
- λ_i can be determined by the condition that the resulting solution x satisfies all m equality constraints

$$g_i(x) = 0, \quad 1 \le i \le m.$$

13. Algorithms (cont-d)

- Thus, equality-type constraint optimization problem can be reduced to an unconstrained one.
- For such problems, many efficient optimization algorithms are known.
- If some of the constraints are inequalities, then the constrained maximum is attained when:
  - some of them are equalities, and
  - some are not.
- In this case, we need to consider all 2^n possible subsets $I \subseteq \{1, \ldots, n\}$.
- For each of these subsets, look for local maxima of the auxiliary function

$$f(x) + \sum_{i=1}^{m} \lambda_i(x) \cdot g_i(x) + \sum_{j \in I} \lambda'_j \cdot h_j(x).$$

14. Algorithms (cont-d)

- For each I, we select the coefficients λ_i and λ'_j from the condition that

$$g_i(x) = 0 \text{ for all } i \text{ and } h_j(x) = 0 \text{ for all } j \in I.$$

- Then, we check that $h_j(x) > 0$ for all $j \notin I$.
- This procedure leads to several different possible maxima x.
- Out of them, we select the one for which the value of the objective function $f(x)$ is the largest.

15. Need for Imprecise ("Fuzzy") Constraints

- In many practical situations, constraints are imprecise.
- For example, when we select a hotel, we want it to be "comfortable" and/or "not very expensive".
- These are not precise terms: in many cases, we are not 100% sure what "not very expensive" means.

16. Fuzzy Logic: a Way to Describe Imprecise ("Fuzzy") Constraints

- Often, we have information described in such imprecise ("fuzzy") natural-language terms.
- We want to take this imprecise information into account when using computers.
- Thus, it is necessary to describe this information in precise terms.
- Such a description was proposed by Lotfi Zadeh and is now known as *fuzzy logic*.
- The main idea behind fuzzy logic is that to describe an imprecise property like "not very expensive":
  - we ask an expert, for each possible value of the corresponding quantity q (i.e., of the hotel rate)
  - to describe, on a scale from 0 to 1, the degree to which this amount is not very expensive.

17. Fuzzy Logic (cont-d)

- For example, for a hotel in El Paso, Texas, a daily rate of \$140 would definitely not satisfy this property.
- So, we assign degree 0.
- The daily rate of \$35 would definitely satisfy this property, so we assign degree 1.
- For some intermediate values like \$80, we will assign intermediate degrees.
- Instead of using the scale from 0 to 1:
  - we can alternatively use a scale, e.g., from 0 to 10, and then
  - divide the result by 10.
- For example, if an expert estimates his/her degree as 7 on a scale from 0 to 10, we get the degree $7/10 = 0.7$.

18. Fuzzy Logic (cont-d)

- As a result, as a description of the desired imprecise property, we get a function that assigns:
  - to each possible value q of this quantity,
  - the degree $\mu(q)$ to which this value satisfies this property.
- This function is known as a *membership function*, or, alternatively, as a *fuzzy set*.
- Tiny changes in x usually only slightly change the degree to which x is possible; so:
  - similarly to our conclusion that the objective function be continuous and even smooth,
  - we conclude that the membership function should also be continuous – and, if possible, smooth.

19. “And”-Operations

- In most practical situations, we have several different constraints that describe different quantities.
- For example, when selecting a hotel, we want a hotel which is:
  - not very expensive (which is a limitation on the daily rate),
  - not very noisy (which is a restriction on noise level), not
  - too far from the city center (which is a restriction on the distance), etc.
- By using the above procedure, we can find:
  - for each of the related quantities,
  - the degree to which the given value of this quantity satisfies the corresponding constraint.

Home Page
Title Page
Page 20 of 47
Go Back
Full Screen
Close
Quit

20. “And”-Operations (cont-d)

- But what we are interested in is the degree to which the hotel as a whole satisfies all these properties.
- We want the degree to which the hotel is not very expensive *and* not very noisy *and* not too far away.
- How can we find this degree?
- Theoretically, we could ask the expert about all possible combinations of values of the corresponding quantities.
- However, the number of such combinations grows exponentially with the number of quantities.
- Even for reasonable number of quantities, the number of queries becomes astronomically large.
- It is therefore not practically possible to ask for expert's opinion about all these combinations.

21. "And"-Operations (cont-d)

- Thus, we need to be able:
  - to estimate the degree to which an "and"-combination $A \,\&\, B$ is satisfied
  - if we know to what extent A and B are satisfied.
- In other words, we need to be able,
  - given the degree a to which A is satisfied and the degree b to which b is satisfied,
  - to come up with an estimate for the degree to which the combination $A \,\&\, B$ is satisfied.
- This estimate is usually denoted by $f_\&(a, b)$.
- The corresponding function $f_\&$ is known as an *"and"-operation*, or, for historical reasons, a *t-norm*.
- This operation must satisfy several reasonable conditions.

22. "And"-Operations (cont-d)

- For example, $A \,\&\, B$ means the same as $B \,\&\, A$.
- So, it is reasonable to require that the estimates for these two combinations are the same, i.e., that
$$f_\&(a,b) = f_\&(b,a).$$
- In mathematical terms, this means that the "and"-operations be commutative.
- Similarly, $A \,\&\, (B \,\&\, C)$ means the same as $(A \,\&\, B) \,\&\, C$.
- So, we expect that the corresponding estimates are equal, i.e., that $f_\&(a, f_\&(b,c)) = f_\&(f_\&(a,b),c)$.
- So, the "and"-operations must be associative.
- The degree to which we believe that $A \,\&\, B$ holds cannot exceed the degree to which A or B holds.
- So, we must have $f_\&(a,b) \le a$ and $f_\&(a,b) \le b$.

23. "And"-Operations (cont-d)

- If our degree of belief in A and/or in B increases, then the degree of belief in $A \,\&\, B$ cannot decrease.
- So, the "and"-operation should be monotonic: if $a \le a'$ and $b \le b'$, then $f_\&(a,b) \le f_\&(a',b')$.
- It is also reasonable, since $A \,\&\, A$ means the same as A, to require that $f_\&(a,a) = a$ for all a (idempotent).
- It turns out that the only monotonic idempotent "and"-operation with $f_\&(a,b) \le a, b$ is $f_\&(a,b) = \min(a,b)$.
- Note that this result does not require that $f_\&$ be commutative and/or associative.
- Minimum is indeed one of the most widely use "and"-operations.

24. Fuzzy Constraints

- So:
  - we apply the appropriate "and"-operation to constraints describing individual quantities, and
  - we get a membership function (fuzzy set) $\mu(x)$.
- This function describes:
  - for each alternative x,
  - to what extend this alternative satisfies all the given constraints,
  - i.e., to what extent this alternative x is possible.

25. Alpha-Cuts: an Alternative Way of Describing Fuzzy Sets

- Instead of a membership function $\mu(x)$, we can describe the same imprecise information if we describe:
  - for each $\alpha \in (0, 1]$,
  - the set $S_\alpha \stackrel{\text{def}}{=} \{x : \mu(x) \geq \alpha\}$ of all the alternatives for which $\mu(x) \geq \alpha$.
- Such sets are known as *alpha-cuts*.
- Once we know all the α-cuts, we can uniquely reconstruct the membership function.
- Namely, $\mu(x)$ is the largest α for which $x \in S_\alpha$.

26. Alpha-Cuts Are Usually Closed and Compact

- Since the membership functions are continuous, alpha-cuts are closed sets.
- Since the set of possible alternatives is bounded, each α-cut is a compact set.

27. Additional Natural Property

- We argued that $f(x)$ and $\mu(x)$ are continuous.
- Similarly, it is reasonable to assume that the set S_α to also continuously depend on α.
- For example, continuous in terms of the usual Hausdorff metric

$$d_H(S, S') = \max\left(\max_{s\in S} d(s, S'), \max_{s'\in S'} d(s', S)\right).$$

- Here the distance $d(s', S)$ between an element s' and a set S is defined in the usual way $d(s', S) \stackrel{\text{def}}{=} \min_{s\in S} d(s', s)$.
- This continuity holds for many known membership functions.

28. Optimization under Fuzzy Constraints: Bellman-Zadeh Formulation of the Problem

- Intuitively, it makes sense to say that the desired alternative should be optimal *and* satisfy all the constraints.
- For example:
  - when we look for a hotel which is the cheapest among all the hotel which are not too far away,
  - what we are really meaning is that we are looking for a hotel which is cheap and not too far away.
- To describe this idea in precise terms, we need to be able to describe:
  - for each alternative x, the degree $\mu_{\text{opt}}(x)$
  - to what extent this alternative is optimal.
- The corresponding degree depends on the value of the objective function $f(x)$.

29. Optimization with Fuzzy Constraints (cont-d)

- So, we must have $\mu_{\text{opt}}(x) = F(f(x))$ for some $F(x)$.
- Let X denote the set of all possible alternatives;
  - when the value $f(x)$ is the smallest possible
  $$f(x) = m \stackrel{\text{def}}{=} \min_{x \in X} f(x),$$
  - then this degree is 0: $\mu_{\text{opt}}(x) = 0$.
- In other words, we must have $F(m) = 0$.
- When the value $f(x)$ is the largest possible, i.e., $f(x) = M \stackrel{\text{def}}{=} \max\limits_{x \in X} f(x)$, then this degree is 1: $\mu_{\text{opt}}(x) = 1$.
- In other words, we must have $F(M) = 1$.
- It is reasonable to use the simplest linear interpolation to find the values $F(f(x))$ for $f(x) \in (m, M)$.

30. Optimization with Fuzzy Constraints (cont-d)

- Thus, we get $\mu_{\rm opt}(x) = \dfrac{f(x) - m}{M - m}$.
- The degree to which the alternative x is optimal *and* satisfies the constraints can be obtained:
  - by applying the corresponding "and"-operation
  - which we agreed to be min, so:

$$J(x) \stackrel{\rm def}{=} \min\left(\frac{f(x) - m}{M - m}, \mu(x)\right).$$

- We should select the alternative for which this degree of satisfaction is the largest.
- So, we should select the alternative for which $J(x)$ attains the largest possible value.
- This formulation was proposed by L. Zadeh and R. Bellman in 1970.

31. Optimization with Fuzzy Constraints (cont-d)

- It is known as Bellman-Zadeh approach to optimization under fuzzy constraints.
- Since both $f(x)$ and $\mu(x)$ are continuous, the function $J(x)$ is also continuous.
- The set of all possible alternatives S is a compact set.
- Thus, there always exists an alternative at which the new objective function $J(x)$ attains its maximum.
- So, the Bellman-Zadeh formulation always leads to a solution.

32. Need for New Algorithms

- At first glance, the situation is good:
  - we have reduced the original practical problem to the problem of unconstrained optimization, and
  - for this problem, as we have mentioned, there are many efficient algorithms.
- However, from the computational viewpoint, the situation is not so good:
  - most efficient optimization algorithms require that the objective function be smooth, and
  - the expression $J(x)$ is not differentiable, even when $f(x)$ and $\mu(x)$ are differentiable,
  - since $\min(a, b)$ is not differentiable when $a = b$.
- So, we need new algorithms.

33. A Known Heuristic Algorithm

- This algorithm assumes that we can efficiently solve the corresponding crisp optimization problems.
- Specifically, we assume that for every α, we can efficiently maximize $f(x)$ under the constraint $x \in S_\alpha$.
- We start with an arbitrary value $\alpha_0 \in (0,1)$, and then compute the values $\alpha_1, \alpha_2, \ldots$ as follows.
- Once we have the value α_k, we:
  - solve the corresponding constraint optimization problem,
  - i.e., we find the maximum M_k of the original objective function $f(x)$ under the constraint $x \in S_{\alpha_k}$
  - and we find the value x_k at which this maximum is attained);
  - then, we compute $\alpha_{k+1} \stackrel{\text{def}}{=} \dfrac{M_k - m}{M - m}$.

34. A Known Heuristic Algorithm (cont-d)

- We stop when the difference between the two consecutive values of α_k becomes sufficiently small, i.e., when

$$|\alpha_{k+1} - \alpha_k| \le \varepsilon \text{ for some } \varepsilon > 0.$$

- In this case, we return the corresponding alternative x_k as the optimal one.
- The main advantage of this heuristic algorithm comes from the fact that:
  - for each α,
  - the constraint $x \in S_\alpha$ has a traditional non-fuzzy form.
- Thus, to find x_k, we can use known efficient algorithms for (non-fuzzy) constraint optimization.

35. Results of Testing This Algorithm

- We applied this heuristic algorithm to several different instances of optimization under fuzzy constraints.
- In all these instances:
  - no matter what the initial value α_0 we selected,
  - the above iterative process converged and let to the solution of the Bellman-Zadeh problem.

36. Theoretical Challenges

- These empirical results led to the following theoretical challenges:
- If the above process converges, do we always get the solution to the Bellman-Zadeh problem (and if yes, why)?
- Why the resulting limit $\lim\limits_k \alpha_k$ did not depend on the initial value α_0?
- Does the above process always converge? and
- If the process does not always converge, how can we modify this algorithm to guarantee convergence?
- In this talk, we provide answers to all these questions.

37. Description in Precise Terms

- Let X be a compact set.
- By a *reasonable membership function* on X, we mean a continuous $\mu : X \to [0,1]$ for which such that $\alpha \to S_\alpha \stackrel{\text{def}}{=} \{x : \mu(x) \geq \alpha\}$ is continuous for $\alpha > 0$.
- By a problem of *optimization under fuzzy constraints*, we mean the triple $\langle X, f, \mu \rangle$, where:
  - $X \subseteq R^N$ is a bounded closed (compact) set,
  - $f : X \to \mathbb{R}$, and
  - μ is a reasonable membership function on X.
- We say that an element x_{opt} is a *solution* to the problem of optimization under fuzzy constraints if

$$J(x_{\text{opt}}) = \max_{x \in X} J(x), \text{ where } J(x) \stackrel{\text{def}}{=} \min\left(\frac{f(x) - m}{M - m}, \mu(x)\right).$$

38. Simplifying Notations

- To analyze our problem, it is useful to reformulate it by using simpler notations.
- For every α, let us denote

$$M(\alpha) \stackrel{\text{def}}{=} \max\{f(x) : x \in S_\alpha\}, \text{ and}$$

$$G(\alpha) \stackrel{\text{def}}{=} \frac{M(\alpha) - m}{M - m}.$$

- In terms of the function $G(\alpha)$, the existing algorithm takes a very simple form: $\alpha_{k+1} = G(\alpha_k)$.

39. Properties of the Newly Defined Functions

- Since $\alpha < \alpha'$ implies $S_\alpha \supseteq S_{\alpha'}$, we have $M(\alpha) \ge M(\alpha')$ and thus, $G(\alpha) \ge G(\alpha')$.
- So, the functions $M(\alpha)$ and $G(\alpha)$ are (non-strictly) decreasing functions;
  - since S_α continuously depends on α and the function $f(x)$ is continuous,
  - one can show that the function $M(\alpha)$ also continuously depends on α.
- Thus, the function $G(\alpha) = \dfrac{M(\alpha) - m}{M - m}$ is also continuous.

40. What Happens When the Process Converges

- When the process converges, i.e., when $\alpha_k \to \alpha$, then:
  - due to the continuity of the function $G(\alpha)$, in the limit $k \to \infty$,
  - we get $G(\alpha) = \alpha$.

41. Answer to the First Challenge

- *Challenge:* why if $G(\alpha) = \alpha$, we solve optimization under fuzzy constraint?
- *Our result:* if $G(\alpha) = \alpha$, then there exists an optimal solution x_{opt} for which $\mu(x_{\text{opt}}) = J(x_{\text{opt}}) = \alpha$.
- Vice versa:
  - for every problem of optimization under fuzzy constraints,
  - there exists an optimal solution x_{opt} for which, for $\alpha \stackrel{\text{def}}{=} \mu(x_{\text{opt}})$, we have $G(\alpha) = \alpha = J(x_{\text{opt}})$.

42. Answer to the Second Challenge

- *Challenge:* why the limit $\lim\limits_{k} \alpha_k$ did not depend on the initial value α_0.
- *Our result:* when the process converges, the limit value α is equal to $\alpha = \max\limits_{x \in X} J(x)$.

43. Answer to the Third Challenge

- *Challenge:* does the process always converge?
- *Our answer: no.* For $X = [0, 1]$, $f(x) = x$, and $\mu(x) = 1 - x$, we have $m = 0$ and $M = 1$, so
$$J(x) = \min(f(x), \mu(x)) = \min(x, 1 - x).$$
- $J(x)$ increases for $x \le 0.5$ and decreases for $x \ge 0.5$.
- So, its largest possible value is attained for $x = 0.5$ and is equal to 0.5.
- Here, for any α, we have
$$S_\alpha = \{x : 1 - x \ge \alpha\} = [0, 1 - \alpha].$$
- The largest possible value $M(\alpha)$ of $f(x) = x$ on this α-cut interval is equal to $1 - \alpha$.
- Since $m = 0$ and $M = 1$, we have
$$G(\alpha) = M(\alpha) = 1 - \alpha.$$

44. Answer to the Third Challenge (cont-d)

- Thus, whatever value $\alpha_0 \le 0.5$ we start with, we get:
  - first, $\alpha_1 = G(\alpha_0) = 1 - \alpha_0$ and
  - then $\alpha_2 = 1 - \alpha_1 = 1 - (1 - \alpha_0) = \alpha_0$ again.
- The iterative process oscillates between α_0 and $1 - \alpha_0$ and does not converge.

45. Answer to the Fourth Challenge

- *Challenge:* guarantee convergence.
- *Our solution:* bisection.
- Start with an arbitrary α_0 and compute $\alpha_1 = G(\alpha_0)$.
- If $\alpha_1 = \alpha_0$, we are done.
- Otherwise we form an interval $[\underline{\alpha}, \overline{\alpha}]$ for which $H(\underline{\alpha}) \geq 0 \geq H(\overline{\alpha})$: $\underline{\alpha} = \min(\alpha_0, \alpha_1)$, $\overline{\alpha} = \max(\alpha_0, \alpha_1)$.
- On each iteration, we take $m \stackrel{\text{def}}{=} \dfrac{\underline{\alpha} + \overline{\alpha}}{2}$ and compute
$$H(m) = G(m) - m.$$
- If $H(m) \geq 0$, we replace $\underline{\alpha}$ with m.
- Otherwise, we replace $\overline{\alpha}$ with m.
- In both cases, the size of the intervals halves.
- We stop when $\overline{\alpha} - \underline{\alpha} \leq \varepsilon$.

46. Acknowledgments

This work was supported in part by US National Science Foundation grant HRD-1242122 (Cyber-ShARE Center).